A&K

1e jaargang no. 5 - 1987
f. 4,95 - BF 98

INFORMATICA

Innovatief



IBM's offensief: MS-DOS, wat nu?

Videotex: commercieel haalbaar?

Cijferagenda: gemak en overzicht voor docent en leerling

Autosimulatie per computer

Kogelbanen

In het volgende nummer van A&K-INFORMATICA starten wij een zesdelige cursus

Werken en experimenteren met MACHINETAAL op MS-DOS computers

Machinetaal is de meest primitieve manier om een computer te programmeren. Met machinetaal kunnen we volledig binnendringen in de ingewanden van de computer. De cursus laat zien hoe we met machinetaal kunnen gaan snuffelen in de kant-en-klare programma's die voor de computer beschikbaar zijn. Machinetaalprogramma's kunnen heel snel werken.

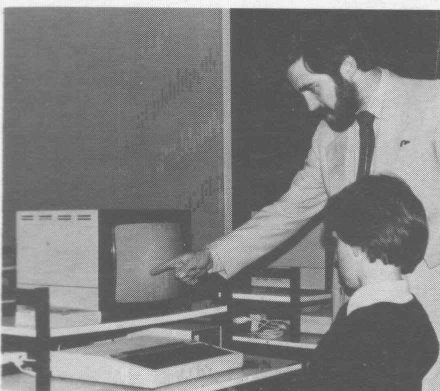
Let dus op de komende nummers van A&K-INFORMATICA

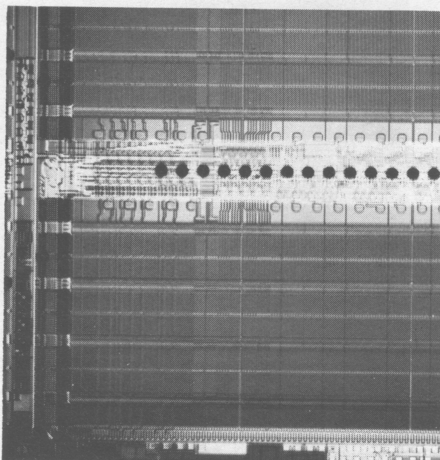
Beter: neem een abonnement op A&K-INFORMATICA, dat geeft de zekerheid geen enkel nummer te hoeven missen. Om het bedrag dat een abonnement kost hoeft u het niet te laten: slechts 39,50 voor een heel jaar (8 nummers boordevol echte informatica en artikelen van de meest uiteenlopende onderwerpen, compleet met listings).

Bel even met 06-0224222 (gratis). Ook gratis is een kaartje naar antwoordnummer 108, 1270 VB Huizen.

INHOUD



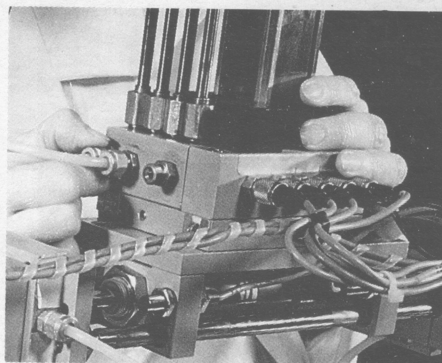




NOS-HOBBYSCOOP

De meest van de in dit nummer geplaatste programma's zullen door NOS-HOBBYSCOOP weer worden uitgezonden. Uitzendingen op zondagen om 22.40 u. en op woensdagen om 19.02 u. Radio 5 - 1008 KHz respectievelijk radio1/2 - FM.





A&K-INFORMATICA wordt geredigeerd door de stichting Mens en Wetenschap t.b.v. ondernemers en bedrijf en voor het onderwijs.
Redactie-adres: postbus 108 - 1270 AC Huizen-NH. Tel.02152-58388

HOOFDREDACTIE: A.C.Sabelis.
EINDREDACTIE: R.Lambert

REDACTEUREN: Dr.P.van Tend, drs.H.Eggen, R.van Dongen, D.Vos.

VORMGEVING: stichting Mens en Wetenschap.

DISTRIBUTIE boekhandel: Betapress - Gilze. Tel.01615-2900.

ADVERTENTIES: Intercomm Publishing - Huizen. Tel.02152-54690

ABONNEMENTEN: per jaar 39,--.
Opgaven: stichting Media Publieksvoorlichting en Onderwijs, postbus 168 - 1270 AD Huizen-Nh.

DRUK: N.D.B. Zoeterwoude.

LITHOGRAFIE: Reproscan - Meppel.

A&K-INFORMATICA verschijnt 8x per jaar.

MS-DOS, wat nu?

IBM is gekomen met een nieuwe reeks personal computers, de Personal System/2 familie.De eerdere personal computers van IBM hebben de norm gesteld voor zakelijke mikrocomputers in het algemeen. Het is de vraag of het Personal System/2 dat weer gaat doen.Nog meer dringt de vraag zich op, wat de toekomst is voor de oude norm, het MS-DOS besturingssysteem, waarin zovelen vertrouwen gesteld hebben.

In het jaar 1981 bouwde IBM zijn eerste personal computers met een 8088 mikroprocessor. Die 8088 chip wordt gemaakt door Intel, overigens een dochteronderneming van IBM. Een van de eigenschappen van de 8088 is, dat hij 16 bits tegelijk kan behandelen. Veel mikrocomputers die in die tijd op kantoren werden ingezet, werkten met slechts 8 bits tegelijk. Zestien in plaats van acht bits lijkt twee maal zo goed, maar met dat soort vergelijkingen is het oppassen. Een letter of een cijfer neemt acht bits in; acht bits noemt men één byte. Wanneer we op een toetsenbord een toets aanslaan, manipuleren we daarmee acht bits. Een 8088 hoeft maar voor de helft in actie te komen. Waarom dan toch een 16-bit mikroprocessor? De voordelen van meer bits ineens komen wel naar voren wanneer hele rijen tekens verplaatst moeten worden. Dat gebeurt bijvoorbeeld, wanneer het beeldscherm van een computer vol is en de hele tekst een regel naar boven moet. Het opschuiven van het beeld noemt men scrollen. Verplaatsingen van series tekens gaan sneller met twee tegelijk, dan stuk voor stuk. Op het ogenblik is de 16-bit (2-byte) mikroprocessor al weer overtroffen. Er is nu de Intel 80386 processor, die 32 bits (4 bytes) tegelijk kan bewerken. Hij doet dat echter alleen als hij er opdracht toe krijgt. Daarin zit het breekpunt dat leidde tot het Personal System/2. Handelingen als het scrollen van het beeldscherm worden in computers uitgevoerd door het zogeheten besturingssysteem. Dit is een programma dat de baas is over alle andere programmma's, die we zelf draaien. Voor het soort mikrocomputers dat IBM maakte, was het besturingssysteem MS-DOS. MS-DOS was gemaakt voor de 8088.Een officiële versie voor de 80386 is er nooit gekomen. Veel van de mogelijkheden daarvan bleven dus ongebruikt. Met Personal System/2 is er nu een besturingssysteem dat ook de 80386 kan uitbuiten. Dat systeem ligt echter niet rechtstreeks in het verlengde van MS-DOS. Die breuk heeft alles te maken met de concurrentie op de computermarkt.

Klonen

MS-DOS is het niet alleen een stuk software. Er worden ook allerlei eisen gesteld aan de opbouw van de hardware. De beschrijving van dit alles was openbaar. Iedereen kon de computers van IBM nabouwen en iedereen kon alle informatie krijgen, die nodig was om er software voor te maken. Beide dingen zijn dan ook op grote schaal gebeurd. Dat er veel software geschreven werd, was in het voordeel van IBM. Die software maakte de IBM PC populair. Het nabouwen door anderen ging IBM echter lelijk opbreken. De nagebouwde MS-DOS computers worden klonen genoemd. Dat is een term uit de biologie. Een kloon is een zelfstandig organisme dat ontstaat uit precies hetzelfde erfelijk materiaal als zijn voorvader of zijn broers. Stekken en pootaardappels zijn voorbeelden van vermenigvuldiging via klonen. De klinkende naam IBM gaf het startsein voor de opmars van MS-DOS. De vele software die ontstond, versterkte het effect. De helft van alle mikrocomputers verkocht in 1986 is MS-DOS. Omdat iedereen klonen



De eenvoudigste computer uit de Personal System/2 familie is het model 30, uitgerust met een 8086 mikroprocessor op 8 MHz. De nieuwe computers zijn iets kleiner in afmeting dan de eerdere PC's.



Het model 50 heeft een 80286 processor, dezelfde die zat in de oude IBM AT mikrocomputer.

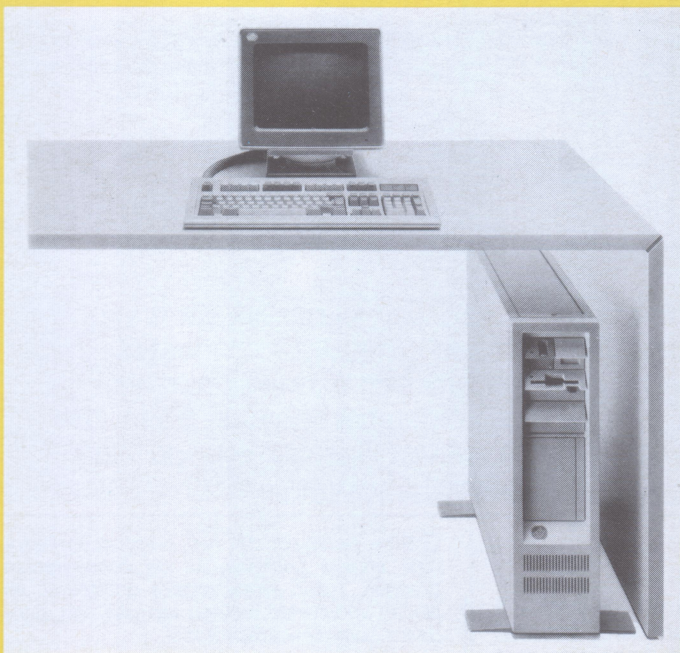
kon bouwen, kwam die 50 procent maar voor een deel ten goede aan IBM. IBM heeft 22.5 procent, de klonenbouwers samen hebben 27.5 procent.
IBM heeft nu geprobeerd het naklonen van het Personal System/2 tegen te gaan. Er zitten bepaalde onontbeerlijke chips in, die via patenten beschermd zijn.In het algemeen zal het inwendige van Personal System/2 niet zo vrij toegankelijk zijn als dat van de oude PC's. Is dat nou een heel slimme zet van IBM of toch niet zo? Alleen de tijd zal het kunnen leren.
Er zullen altijd mensen zijn, die niet anders durven kopen dan IBM. Een zekere basismarkt is zo dus verzekerd. Bij MS-DOS boden de klonen meer voor minder geld.Het Personal System/2 is niet meer zo extreem duur als vroeger de oude produkten van IBM, naar prestatie geschat. Het basismodel is f. 4749,-.exclusief B.T.W. Het meest luxueuze lid van de familie komt op f. 24.876,-.

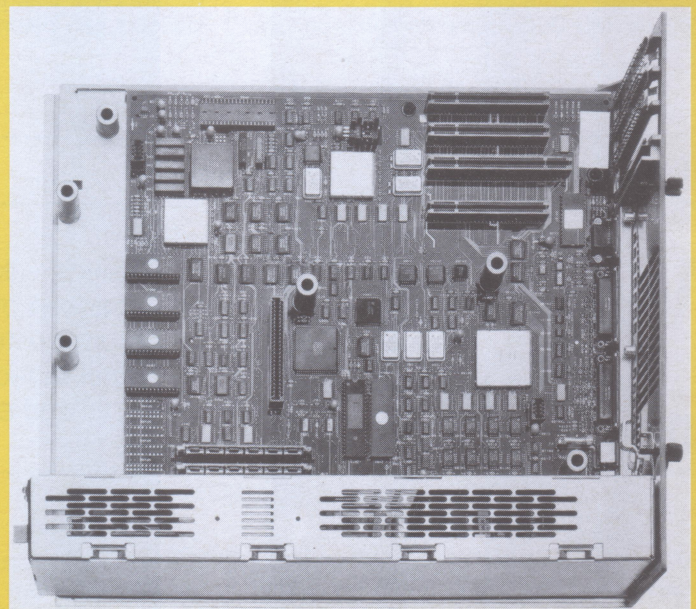
Veranderingen in hardware

De Personal System/2 computers kunnen op het ogenblik nog steeds MS-DOS programma's verwerken. Een probleem is echter dat het Personal System/2 gebruik maakt van 3.5 inch diskettes. De meeste MS-DOS machines werken met 5.25 inch diskettes. Zonder hulpstukken kunnen ze elkaars diskettes dus niet lezen. De 3.5 inch diskettes zijn niet de enige verandering in de hardware. De grafische mogelijkheden van het beeldscherm zijn verbeterd. De grafische mogelijkheden van de oude IBM PC's waren dan ook wel erg mager ten opzichte van wat technisch mogelijk was. Allerlei kloonfabrikanten hadden op dat punt al verbeteringen aangebracht in hun produkten.
Het basismodel van het Personal System/2 (model 30) werkt met een 8086 mikroprocessor. De 8086 verschilt maar een heel klein beetje van de 8088 van de oude PC. De 8086 kwam op de markt in 1978. Hij is dus nogal oud en was dat eigenlijk ook al toen in 1981 de eerste PC's verschenen. Wel gebruikt men op het ogenblik een versie die sneller kan lopen. Het is niet zomaar conservatisme dat IBM werkt met oude mikroprocessortypes. Een mikroprocessor wordt door de computergebruiker gelukkig meestal ervaren als onfeilbaar. Toch blijkt een nieuwe mikroprocessor na het uitkomen vaak nog ontwerpfouten te bevatten. Ook bij de 8086 was dat zo. Verder moet een computerfabrikant verzekerd zijn van een gestage aanvoer van chips. Ook daaraan schort in het begin nog wel eens het een en ander. Het is dus nog niet zo gek te werken met een chip die zich al jarenlang bewezen heeft. In dat licht is het voor IBM wel haast een waagstuk nu al een 80386 mikroprocessor toe te passen in de topmodellen van de reeks. De kloonfabrikanten zijn nog geen jaar met die chip bezig! De klonen moeten het op de 80386 doen met MS-DOS, dus met software die niet ten volle gebruik maakt van de grote 32-bits-stappen die de 80386 kan zetten. Wel profiteren ze van het feit dat de 80386 heel veel stappen per seconde kan zetten.



Tot de Personal System/2 familie hoort ook een optische schijfeenheid met een capaciteit van maar liefst 200 megabyte per schijf. De verwisselbare schijven zijn eenmaal te beschrijven en daarna vele malen uitleesbaar. Dit heet WORM: Write Once Read Many.


Het model 80 is met een 80386 processor de snelste van de Personal System/2 familie.Zijn vaste schijf heeft een opslagcapaciteit van maximaal 230 megabyte. Op het ogenblik is 20 of 40 megabyte gangbaar.


Op deze systeemplaat van een model 50 zijn allerlei dingen ingebouwd, die vroeger los bijgeplaatst moesten worden, bijvoorbeeld voorzieningen voor communicatie met andere computers.

MS-DOS voorlopig de baas

MS-DOS is de baas over de gebruiksprogramma's die op een computer draaien. Op een 8086 heeft MS-DOS echter geen werkelijke mogelijkheden om de baas te spelen. Op de 80386 zijn er wel voorzieningen die een besturingssysteem speciale bevoegdheden geven, met name bij het indelen van het geheugen. Het Personal System/2 maakt ook werkelijk gebruik van die mogelijkheden. Er kunnen dan schijnbaar meerdere programma's tegelijk draaien. In feite wisselen die programma's elkaar af onder toezicht van het besturingssysteem. De makers van de programma's hoeven niet speciaal rekening te houden met een eventuele afwisseling door andere programma's. Alle vervelende regeltaakjes die daarbij komen kijken, worden overgenomen door het besturingssysteem. Bij MS-DOS zijn zulke voorzieningen er niet. MS-DOS kon een maximum van 640K aan geheugen aan. Naarmate er steeds omvangrijker software op de markt kwam en computers met grotere geheugens verschenen, werd die beperking nogal voelbaar.Er zijn vele mogelijkheden om verder te komen dan 640K en die zijn dan ook ijverig toegepast. Het

probleem is dat de verschillende systemen die er nu zijn, de uitwisseling van programma's bemoeilijken. Wat geheugenomvang betreft kan het Personal System/2 een grootte-orde verder denken dan MS-DOS. De technische documentatie van het Personal System/2 zal, zoals gezegd, niet zo vrij verkrijgbaar zijn als die van MS-DOS. Die vrije verkrijgbaarheid heeft overigens nooit zo erg gegolden voor hobbyïsten; over het inwendige van bepaalde andere mikrocomputers is veel gemakkelijker informatie te krijgen.
De sluier rondom het Personal System/2 zal namaak voorkomen, maar ook het ontstaan van software remmen. Dezelfde situatie heeft bestaan rond de Apple Macintosh computer. Deze nam vorig jaar 8 procent in van de mikrocomputermarkt. Hij is echter in opkomst. November 1986 was de eerste maand waarin meer Macintoshes werden verkocht dan IBM's. Voor nieuwe Macintosh uitvoeringen heeft Apple nu wel de technische informatie vrijgegeven. De opmars kan doorzetten.
IBM zoekt bescherming achter een sluier, Apple haalt juist de schil eraf. Maar er zijn ook nog de kloonfabrikanten. Zij zouden zich kunnen gaan wijden aan het namaken van Macintoshes. Waarschijnlijk zal dat pas lonen als de echte Mac (dank zij nieuwe software) wat meer is ingeburgerd. Kloonfabrikanten kunnen voorlopig beter bij de MS-DOS computers blijven. De Volkswagen Kever is ook nog jarenlang in Mexico geproduceerd, nadat de Europese fabricage was gestaakt. Wanneer de kloneurs een betere versie van MS-DOS algemeen aanvaard zouden weten te krijgen, heeft IBM toch nog het nakijken.



Computers uit de Personal System/2 familie kunnen met elkaar verbonden worden tot een zogeheten local area network (LAN).

□

D. Vos

Computers

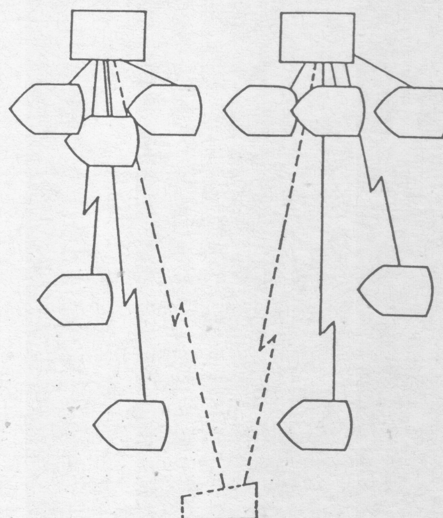
VIDEOTEX

Deel 1

Een steeds vaker opkomend woord wanneer er sprake is van databanken of computernetwerken is het woord 'Videotex'. Daarom besteden we in enkele artikelen aandacht aan het verschijnsel Videotex. Aan de orde komen het ontstaan ervan, hoe videotexnetwerken groeien en het nut van videotex voor de 'eind'-gebruiker.

Videotex is een tiental jaren geleden bedacht door de Engelse PTT om (schrik niet) de kosten van de telefoonlijnen omhoog te brengen. De PTT had n.l. ontdekt dat er sprake is van twee pieken in het telefoonverkeer, en zocht naar een middel om de 'slappe' perioden in de rest van de dag in te vullen.
Hierom werd gedacht aan een manier om Teletekst-achtige beelden per telefoonlijn over te seinen naar kleine homecomputers van telefoon-abonnees. Er werd een standaard 'protocol' ontwikkeld (manier om gegevens over te zenden) en tevens een standaard indeling van de over te zenden 'pagina's'. Dit protocol kennen we vandaag de dag nog als het PRESTEL videotex protocol. Het gezicht van databanken als Girotel is in PRESTEL. Er zijn nog meer faktoren die dit gezicht bepalen, zoals:
① een sterretje onderaan de pagina waar de gebruiker het paginanummer kan aangeven waar hij naar toe wil (de pagina's van informatie zijn genummerd; een wat beter systeem biedt ook benaming met trefwoorden, zodat U door een woord in te tikken naar de desbetreffende pagina kunt springen);
② standaard afmetingen van de scherm pagina's;
③ een standaard wijze van aanloggen en ingeven van passwords (wachtwoorden die dienen voor de beveiliging);
④ het weergeven van keuzemogelijkheden op een pagina door opties 1 tot en met 9, zodat keuzes steeds op dezelfde manier gemaakt kunnen worden;
⑤ in een beter systeem: het bieden van antwoordpagina's, waarin naam, adres en woonplaats ingetikt kunnen worden (wanneer men iets wil bestellen, of informatie wil aanvragen);
⑥ een standaard manier om kleuren weer te geven (in PRESTEL nog heel gebrekkig, later verbeterd met de Vidibord optie

Diverse bedrijven die hun informatienetwerk naar andere bedrijven en de consumenten brengen, zullen elkaar ergens ontmoeten en alsdan voorbereid moeten zijn met een standaard communicatieprotocol.

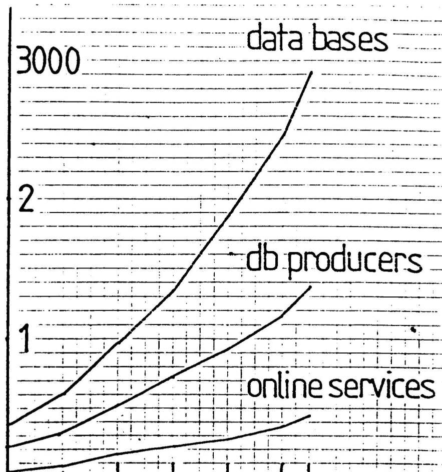


(nooit van de grond gekomen, maar bestaande uit een codering voor scherpe lijnen), en de CEPT standaard, die het hele beeld een hogere resolutie geeft).

Consument gericht

Er valt nog veel meer over de standaard te vertellen, maar een belangrijke eigenschap van het Engelse PTT systeem springt in het oog: het is consumentgericht, net zoals Viditel dat is. Dit betekent dat de PTT gedacht heeft in één keer de massa te kunnen bereiken, onder voorbijgaan van het bedrijfsleven. Deze gedachte is mislukt, en het streefgetal van honderdduizenden abonnees is nooit gehaald. Wel scoort het systeem een matig succes met momenteel ruim 20.000 abonnees. (Interessant is ook dat de Ford Foundation deze grootschalige gedachte in Engeland gesteund heeft met een schenking van tientallen miljoenen dollars.)
Onder deze 20.000 bevinden zich echter aansluitingen van abonnees die betaald worden door hun bedrijf, van bedrijven die een test doen, of gewoon nieuwsgierig zijn, enz.
In Frankrijk heeft de overheid (onder invloed van het 'volkscomputer'-idee) op grote schaal Minitels ter beschikking gesteld aan de bevolking. Dit zijn videotexterminals (eindgebruiker stations) waarmee telefoonnummers kunnen worden geraadpleegd. Er hoeven dus minder telefoonboeken te worden gedrukt.Als belangrijk argument werd opgevoerd dat er dus minder bossen behoefden te worden gekapt. Naast telefoonnummers zijn grote hoeveelheden informatie van bedrijven inmiddels te raadplegen.
In Nederland heeft men tot dusverre de Franse proef niet klakkeloos willen herhalen, maar streeft men naar iets veel mooiers: volledige integratie van tekst, beeld en geluid, en wel in twee richtingen. In elk geval worden diverse proeven gedaan, o.a. in het Limburgse, met zware subsidies van de overheid, om dit ideaal te bereiken. (Hopelijk wordt het een betaalbaar ideaal.)

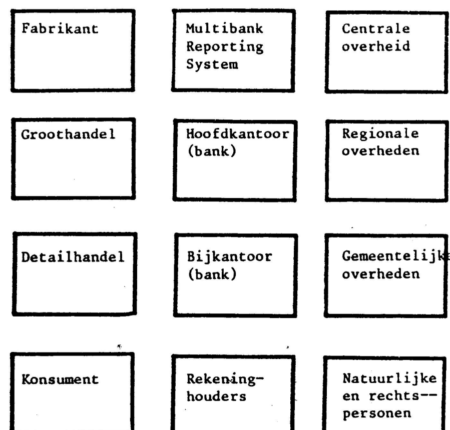
Grafische weergave van de steeds toenemende snelheid waarmee databases en on-line services hun intrede doen (bron: Cuadra Gids). De werkelijke aantallen liggen naar schatting nog een factor twee tot drie hoger.



Wantrouwen consument

Het wantrouwen tot dusverre van de consument is niet geheel ongegrond:
① het systeem is bedacht om de PTT meer omzet op te leveren, en het kost de abonnee dus geld;
② het aantal databanken is (nog) gering, bovendien is de zoekstructuur nog gebrekkig. Als er gezocht kan worden op niet meer dan een sleutelwoord tegelijk, dan zoekt een papieren publicatie (zoals de Gouden Gids) sneller en overzichtelijker. Zoeken op meerdere trefwoorden tegelijk zou een oplossing zijn, maar dit wordt door Viditel niet geboden, en het vergt bovendien veel meer processorcapaciteit (het zou veel meer kosten);
③ het huren van een Viditel terminal kost weer extra geld en het gebruiken van een eigen PC met 'terminal emulatie' (nabootsing van een videotex terminal) geeft weer andere problemen. Er moet een modemaansluiting komen, de baudrate (seinsnelheid) moet worden ingesteld, het modem moet gekozen zijn voor de juiste baudrate (75 en 1200 baud), het pakket dat een Videotex terminal nabootst moet worden geïnstalleerd, e.d.;
④ vanwege de Teletekst achtergrond van Videotex worden steeds hele pagina's (schermen) over gestuurd. Verandert er dus een letter, dan wordt toch het scherm opnieuw opgebouwd. Dit kost veel tijd. Weliswaar is de PTT in staat enkele velden over te zenden (wat veel sneller gaat), maar daar is ze dus niet mee gediend. (De PTT verdient aan de verbindingsduur, bovendien heeft de PTT langere termijnplanningen met Videotex.);
⑤ de beelden zijn nog niet 'mooi'. Amerikanen vinden PRESTEL zelfs afschuwlijk. CEPT is veel scherper en mooier, zonder nadelige gevolgen voor de lijnkosten. Amerika kent o.a. NAPLPS en Japan werkt aan CAPTAIN als standaards die 'beter' moeten zijn;
⑥ de terminal emulatie is niet een-tweedrie te integreren met de eigen programmatuur. Bijvoorbeeld men kan niet met het eigen boekhoudpakket de betalingen middels Girotel aansturen. Het ontbreekt ook aan gemakkelijke middelen om teksten uit een tekstverwerker in een Videotexsysteem als Viditel te zetten.

Middels videotex systemen kan zowel verticale als horizontale integratie plaatshebben binnen en tussen verschillende branches.



Bedrijfsleven

In de gebreken van het Viditel systeem van de PTT is echter voorzien door bedrijven, die zelf centrale Videotex systemen hebben ontwikkeld. Deze hebben nog weinig te maken met een Videotexpakketje op uw PC thuis, maar zijn grote pakketten op mini- en mainframe computers, waarmee bedrijven zelf hun eigen videotexsysteem in huis kunnen halen, tot een aantal malen groter dan wat de PTT in huis heeft met Viditel. (Er zijn koppelingen mogelijk van de Viditelcomputer naar externe videotexsystemen, maar nog niet omgekeerd.) Het gaat de gebruikers van deze systemen voornamenlijk om 'business to business' communicatie. Dit is het bereiken van de decentrale vestigingen vanuit een hoofdkantoor, of verbinding met ambulante vertegenwoordigers. Pas in de tweede plaats gaat het om het bereiken van het publiek.

Waarom Videotex

De vraag is waarom deze bedrijven Videotex hebben aangegrepen voor hun 'business to business' problemen. Dit wordt ook niet gedaan door bedrijven die al een prachtig netwerk hebben naar hun vestigingen (zoals Albert Heijn bijvoorbeeld). Het gaat om bedrijven die nieuwe netwerken aan het opbouwen zijn (eventueel naast bestaande netwerken), zoals banken die honderden bijkantoren ontsluiten (Amro, Rabo, NMB e.a.) met diensten die uiteindelijk ook op het publiek gericht zullen zijn. Er spelen dus twee factoren:
① de noodzaak tot een standaard procotol en wijze van presentatie.De reden hiervoor is dat steeds meer netwerken zich vanuit bedrijven uitbreiden, en dat men van de gebruiker niet meer kan verwachten dat hij verschillende soorten terminals voor verschillende netwerken naast elkaar op zijn bureau zou hebben. Eerder zal de gebruiker eisen dat hij met zijn PC of terminal al die verschillende netwerken op dezelfde manier kan benaderen;
② een verstandige planning, in die zin dat eerst de eigen bedrijfsvestigingen worden 'bediend' en dan pas de consument. De oorspronkelijke fout van de PTT (rechtstreeks naar de consument) wordt zo vermeden, en groeiproblemen kunnen worden opgelost voordat men het publiek bereikt.

Generatie van videotex

Er zijn inmiddels al diverse generaties van (centrale) videotex systemen verschenen:
① de oudste versie, bestaande uit alleen 'platte' tekstpagina's (dit is de versie die inmiddels ook op kleine PC's aangeschaft kan worden, die tot 32 gebruikers kunnen bedienen). De grotere systemen kunnen bijvoorbeeld 1600 gelijktijdige gebruikers (of een veelvoud) bedienen, wat neerkomt op 16.000 abonnees (of veelvoud);
② een nieuwere versie die het mogelijk maakt applicaties (programma toepassingen) aan pagina's te koppelen. Met andere woorden, de (eind) gebruiker kiest een bepaalde pagina, en daardoor start een applicatie op (zoals een boekhoudpak-

ket).Het videotexpakket dient nu tevens als 'supermenu', waaronder alle programmatuur gegroepeerd kan worden;
③ volledige integratie, door applicaties op te starten, die zelf weer inloggen op een andere databank. Eventueel kunnen protocol en zoektaal voor de andere databank door het videotexsysteem worden verzorgd. De wereld komt er voor de eindgebruiker zo eenvoudig en universeel uit te zien.

Onderdeel van deze integratie moet zijn een mogelijkheid om veel gelijktijdige communicatie op te vangen. Dit stelt hoge eisen aan de hardware. Het videotex systeem voor de Londense beurs is bijvoorbeeld speciaal ontwikkeld voor de Classic II mini computer serie van ModComp, vanwege de hoge 'throughput' van de machine (communicatiemogelijkheden, die de Classic II vanouds aan de aluminiumindustrie heeft geboden, daar benodigd voor het aansturen van processen in 'real time').
Het kunnen functioneren als Front End systeem valt hier eveneens onder. Dit is een computer die toegang geeft tot een grotere computer (mainframe) waarvan hij zich als terminal voordoet. Naar de eindgebruiker toe doet een Front End systeem zich in dit geval echter als Videotexsysteem voor.

Kleinere systemen

Het is niet te verwachten dat al deze voordelen zich ook gaan voordoen op kleinere systemen. Daar zijn hele goede redenen voor, die te maken hebben met:
a. benodigde computerpower;
b. het 'aan' kunnen van grote aantallen gebruikers, ZONDER dat hierdoor alle computerpower wordt benut. Met andere woorden, het systeem moet voldoende power overhouden om allerlei programmatuur te kunnen draaien;
c. het op maat schrijven van programmatuur vergt vaak een investering die in buitensporige verhouding zou staan tot zeer goedkope software;
d. de storingsgevoeligheid van deze systemen worden bijzonder cruciaal.

Er zijn wel een tweetal lichtpuntjes, n.l. de opkomst van Xenix en vooral Unix (ook voor PC/ATs en straks de 386 processor) (denk ook aan een groeipad naar grotere mini's en aan de voordelen van de X/Open Group, wat een soort van standaard Unix inhoudt), de mogelijkheid tot networking onder videotex. Door de applicaties in een netwerk op andere machines aan te bieden dan de videotex host, wordt het geheel krachtiger en flexibeler. De kosten en beheersbaarheid van het geheel moeten echter nog worden bezien.

Toepassingen voor de eindgebruiker

We hebben nu de historische ontwikkeling van videotex in vogelvlucht beschreven. Pas nadat bedrijven hun systemen geïnstalleerd hebben en ermee in de lucht zijn, krijgt de eindgebruiker er op grote schaal mee te maken.
De meer technisch aangelegde gebruiker kan zich amuseren met een kleinschalig videotex systeem van de eerste generatie. Zoiets kost enkele duizenden guldens en is een goede testmogelijkheid voor grotere concepten. Commerciële haalbaarheid is echter gering: via een 06-nummer van de PTT kan maximaal f 0,50 per minuut worden geïncasseerd (waarvan slechts de helft resteert). Bovendien: voor welke informatie is de consument bereid zoveel te betalen? Reden waarom grotere bedrijven videotex of gebruiken voor business to business verkeer, of als zakelijke test tegen kostprijs, of als test richting consument onder kostprijs.
De echte eindgebruiker begint nu langzaam iets te merken van de tests die zijn richting uitkomen. Wat hij nodig heeft is een videotex terminal, of PC met emulatiepakket, en een modem, om enkele databanken onder videotex te kunnen 'ontsluiten'. Deze mogelijkheden worden in het volgende artikel in het kort omschreven, alsmede de verder te verwachten ontwikkelingen.

Kleurenmonitor met hoge resolutie



Een nieuwe kleurenmonitor, 14 inch, komt binnenkort op de Nederlandse markt. Het betreft de EIZO FlexScan 8060 geïntroduceert door de firma Manudax te Heeswijk.
Deze monitor is in staat om volledig automatisch op de toegepaste grafische adapterkaart af te stemmen. Daarmee vervalt de noodzaak om voor applicaties, die verschillende kaarten vereisen, afzonderlijke monitoren aan te schaffen.
Bij gebruik van een IBM compatible PC ondersteunt de FlexScan de MDA, HGC, OGA, EGA en PGA standaards., Met zijn dotafstand van slechts 0,28 en zijn hoge resolutie (820 x 620 beeldpunten) heeft de EIZO een aanzienlijk betere beeldkwaliteit dan vergelijkbare monitoren, die een dotafstand hebben van 0,31 mm en een maximale resolutie van 800 x 560 hebben. Door deze faciliteiten is de nieuwe EIZO monitor bij uitstek geschikt voor toepassingen waar hoge eisen aan de beeldkwaliteit worden gesteld, zoals professionele tekstverwerking, CAD(Computer Aided Design) en DTP (Desk Top Publishing)

Portable computer

Het wordt een steeds duidelijkere trend dat vooral zakenlieden de inhoud van hun attachékoffer, die meestal boordevol papieren en andere informatieverschaffende middelen zit, verruilen voor een draagbare PC die in dezelfde koffer past.
Ook de Toshiba fabrikant doet daar mee een



nieuwe portable PC aan mee en wel de T1100-plus. Het is een krachtige, lichtgewicht (4,5 kg) computer die op een 8086 processor draait. Het geheugen bestaat uit 640 Kbyte en twee ingebouwde floppy diskdrives (720 Kbyte geformatteerd). Het scherm bestaat uit een Liquid Cristal Display (LCD-scherm) en is ook bij felle zon duidelijk leesbaar (640x200 pixels).

D. Vos
Siso code 365

Introductie in de computer

Ongeveer het laatste aspect waar de beginnende computergebruiker aan toekomt is dat van datacommunicatie. Dit is het verzenden en ontvangen van gegevens over vaste kabels, maar ook over (tijdelijke) telefoonverbindingen. Vooral dit laatste wordt voor de computergebruiker uiteindelijk het belangrijkst. De maatschappelijke veranderingen door data-communicatie moeten eigenlijk nog gaan komen. Ze zullen in elk geval veel verder strekken dan alleen maar wat berekeningen versnellen, wat aanvankelijk de rol van de computer was. Daarom ontbreekt het onderwerp data-communicatie (D.C.) in deze introduktie niet.

Deel 5 - Data communicatie

Twee toonhoogten

We hebben al gezien dat een 0 bestaat uit een positie van 0 Volt in een chip, en een 1 bestaat uit een positie van 5 Volt (of lagere spanning in recente chips). Begrippen als 'aan' en 'uit' van machines, als 'ja' en 'nee' voor beslissingen kunnen dus worden vertaald in 0/5 Volt. Maar ook getallen kunnen worden omgezet in series posities van 0/1. Het getal 0 als een 0, het getal 1 als een 1, 2 als 10, 3 als 11, 4 als 100, 5 als 101, enzovoorts. Door cijfers getallen toe te kennen (a=65, b=66, c=67 enz.) kunnen hele teksten worden opgeslagen als series van posities van 0/5 Volt. Wat met die gegevens ('data') gebeuren moet, wordt eveneens opgeslagen als 0/1 positie-series; evenals het eindresultaat van bewerkingen.

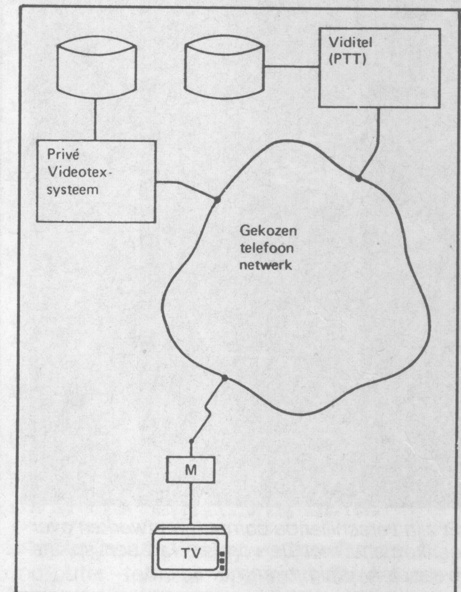
De vraag rijst nu hoe die 0/1 informatie over een telefoonlijn kan worden verzonden. Dit gebeurt door een 0 om te zetten in toonhoogte 1 en een 1 in toonhoogte 2. Het apparaat dat dit bewerkstelligt, heet 'modem' en vertaalt aan de zendende kant de 0/1 in twee tonen, en aan de ontvangende kant de tonen in 0/1 informatie. (Beide kanten kunnen ook tegelijk zenden en ontvangen.) De frequenties van de tonen verschillen ietwat in verschillende werelddelen. Het gaat om duizend tot tweeduizend Hertz. Het herkennen van de tonen gebeurt met filters voor juist die twee frequenties. De tonen liggen dus niet te dicht bij elkaar, anders kunnen de filters geen onderscheid maken.

Verschillende protocols

Wanneer twee apparaten met elkaar 'praten' over een lijn, dus 0/1 informatie uitwisselen, dan zijn er bepaalde 'beleefdheidsafspraken'. (Protocols.)
Bijvoorbeeld: mogen de computers tegelijk aan het woord zijn (duplex), mogen ze zelf bepalen wanneer ze tekens overzenden (asynchroon), moeten ze controlebits meezenden om storingen op te sporen, enzovoorts. Andere zaken zijn: wat zenden de computers elkaar toe (ASCII of andere tekens), hoe lang mogen de treintjes van data zijn (bijvoorbeeld 80 in het X25 protocol). Verschillende protocols door elkaar kunnen ook al verwerkt worden (X400).



Data-communicatie via de telefoon.



Laag boven laag

Het zal duidelijk zijn dat er sprake is van diverse 'lagen' van informatie:
- de vaste of tijdelijke verbinding zelf,
- de toonhoogten die gebruikt worden om de bits (0/1 informatie) over te zenden,
- de manier waarop de bits gegroepeerd worden in bytes e.d., en de afspraken over de series van bytes,
- de afspraken over de tekens die een computer terugstuurt, en hoe een 'remote host' (verre computer die men per telefoon kan bedienen) aangestuurd moet worden.

OSI

Deze afspraken zijn internationaal vastgelegd in de lagen van het z.g. 'OSI-model', wat voor communicatie tussen grotere bedrijven relevant is. Het is echter een nogal formeel model en invoering gaat geld kosten, ook voor grote bedrijven soms te veel geld. Het is bovendien nogal ingewikkeld. (Het monopolie van de PTT over de D.C. wordt beperkt tot de onderste lagen van het OSI model, wat natuurlijk wel de belangrijkste zijn. In wezen verandert er dus weinig aan het monopolie.)

Eenvoudiger protocols

Vrijwel iedereen is wel positief t.o.v. een standaard manier van met elkaar communiceren en het OSI-model (niet de kosten), maar wil ook best iets eenvoudigers. De vraag is wat? Er zijn diverse feitelijke deelstandaards, zoals het XMODEM protocol (onder PC's in de VS), en Videotex, waarvan ook weer onderdelen zijn: Prestel, CEPT, NAPLPS, Captain e.d. Dit zijn allemaal deelprotocols, voor kleine onderdelen van het hele probleem. Wel mag worden verwacht dat door 'survival of the fittest' een van de protocols komt bovendrijven als de meest gangbare (na keiharde concurrentie van de desbetreffende mammoetbedrijven die elk trachten hun standaard als de enige echte naar voren te schuiven).

Netwerken

Er zijn verschillende manieren om computers met elkaar te verbinden. Plaatselijk kunnen kabels gelegd worden om computers te verbinden. Die kabels kunnen dan goed worden ontstoord. Er is ook netwerksoftware nodig, die al of niet als een keurslijf werkt. Zo'n netwerk heeft een LAN.
Een netwerk voor een wat omvangrijker gebied heet WAN, ofwel Wide Area Network. Verschillende leveranciers hebben verschillende netwerk paradepaardjes die zij koste wat kost willen verdedigen, pushen en promoten. En, u raadt het al, ze sluiten niet per se op elkaar aan. Gelukkig zijn er weer brugfuncties tussen de verschillende netwerken. Helaas wordt het geheel hierdoor razend complex, en bijna maandelijks ingewikkelder.



Er zijn verschillende computernetwerken overeenkomend met de vormen van een spinneweb: stervormig, cirkelvormig etc.

RACE netwerk

In Europa, de VS en Japan wordt gewerkt aan het 'RACE' netwerk, een z.g. breedbandig netwerk waarin ruimte is voor de bestaande PTT-lijnen voor spraak, voor data-communicatie en voor videosignalen. Het zal dan (na het jaar 2000) mogelijk zijn naar Japan te bellen, met beeldverbinding en per computer het Japans vertaald te krijgen, en omgekeerd. (Verrassende effecten kunnen daardoor ontstaan, als u weet dat in het Japans het woord 'niet' helemaal aan het eind van een (soms lange) zin staat!) Nu al staat vast dat het RACE netwerk onbetaalbaar duur zal worden. Een soort statussymbool voor managers van de Top 500 bedrijven dus.

Concurrentie

De concurrentie staat echter niet stil. Het idee wordt grondig bestudeerd en dan liefst in sterk versimpelde vorm overgenomen. Zo is het Videotex protocol afkomstig van de Engelse PTT (Prestel), waarmee de PTT twee vliegen in een klap wilde slaan:

- de telefoonlijnen drukker bezet krijgen in slappe uren
- grote belangen dienen en zich daarin een belangrijke marktpositie veroveren. (De Ford Foundation zag die belangen eveneens, en schonk 50 miljoen dollar.)

Het PTT experiment (in Nederland herhaald als Viditel), is in die zin mislukt dat het concept te centraal was en te duur voor de eindgebruiker. In Frankrijk werd dit opgelost door terminals (Minitels) cadeau te doen i.p.v. een telefoonboek. In Nederland wordt nog gestudeerd op een volledig geintegreerde oplossing. Er is echter een duidelijk pluspunt. De PTT heeft internationaal met haar grootschalige opzet een standaard weten te zetten (Prestel en CEPT), die het middenbedrijf heeft kunnen overnemen. Videotex is nu een standaard buiten de PTT om.

Flexibel

Het bedrijfsleven stelt zich daarbij veel flexibeler op dan de PTT. Zo stuurt Viditel alleen hele beelden over wat lang duurt. Begrijpelijk gezien de achtergrond van Teletekst en de geplande groei richting videobeelden onder het RACEnetwerk. Een eigen videotexsysteem (dat een bedrijf



Televisietoestel geschikt gemaakt voor Viditel met behulp van telefoon en voorzetkastje.

zelf in huis heeft), kan echter ook enkele velden overzenden, i.p.v. hele beelden. Hele beelden kunnen worden overgezonden als 'electronic mail' in combinatie met Videotex (wat een continue lijnverbinding vergt). Er kunnen applicaties draaien die gekoppeld zijn aan een Videotex pagina ('frame').

Stromingen

Er zijn diverse stromingen gaande:
① het gebruik van Viditel en Memocom (het PTT elektronisch berichten systeem) voornamelijk als proef en/of door bedrijven in de overheidssfeer,
② het gebruik van een eigen Videotexsysteem om het publiek te bereiken,
③ het gebruik van een eigen Videotexsysteem om de eigen kantoren te bereiken (diverse banken),
④ het blijven gebruiken van gewone ASCII-terminals om de eigen kantoren en vestigingen te gebruiken, vooral als er al fraaie netwerken zijn opgebouwd,
⑤ proberen Videotex-terminal emulatie te verwezenlijken op PC's, vooral als op de PC nog andere programma's kunnen draaien. Het liefst ook nog gelijktijdig, zodat u zowel uw boekhoudpakket draait, als ingelogd bent in Girotel om uw betalingen elektronisch door te geven,
⑥ de poging te werken naar andere standaards, zoals die van IBM, die beter aansluiten op die van mainframes (die IBM graag verkoopt, en die natuurlijk door het opkomende dataverkeer, steeds meer nodig worden).
De overgang van IBM van PC onder PCDOS naar OS/2, is hiervan een voorbode. (En verstoort de MSDOS standaard.),
⑦ pogingen in de VS en Japan om andere, betere Videotex standaards te ontwikkelen,
⑧ netwerken onder Xenix en Unix of simulaties daarvan op PC's,
⑨ de trend om pakketten op verschillende machines over te zetten, zodat hetzelfde pakket in elk geval met zijn broertjes kan praten (dit is een kostbare manier, die voor gevestigde pakketten wel werkt, zowel op machines van dezelfde grootte, als op kleine en grote machines. Bijvoorbeeld: Oracle op PC, RPG II nu ook op PC.



Beeldtelefoon ook een vorm van data-overdracht.

Verwacht mag worden dat deze trend zich uit zal breiden tot D.C.),
⑩ combinaties hiervan, simulaties, of (onwaarschijnlijk) totaal nieuwe ontwikkelingen. In die zin onwaarschijnlijk dat de gevestigde massa van programma's en investeringen nieuwe, andere ontwikkelingen zal afremmen, tenzij de laatste echt fabelachtig zijn.

C en Unix

Het lijkt waarschijnlijk dat Unix van al deze confusie de lachende derde zal worden. Nog is de 'kritische massa' niet bereikt (hoeveelheid installaties waarboven Unix steeds harder gaat lopen, met name in het bedrijfsleven). Deze zal echter wel komen. Het aantal C-programmeurs in de V.S. is sterk aan het toenemen.
Vroeg of laat zal er een standaard uitkomen voor D.C. en deze standaard zal goedkoop en simpel genoeg moeten zijn.

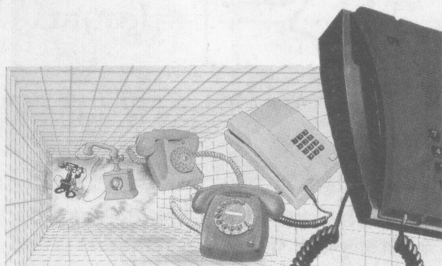
Kennis van data-communicatie

Beter dan afwachten wat over zal komen (en er dan helemaal niets meer van te be-

grijpen), is het om alvast de simpele basisbeginselen van Data-Communicatie onder de knie te krijgen. Dit doet u door diverse modems te bestuderen, diverse D.C.-pakketten (liefst de geintegreerde). Lees eens een boek over D.C.
De zogenaamde RS232c interface is een standaard die men bijvoorbeeld goed kan bestuderen als men iets meer wil weten van D.C. Zie o.a. de technische handleidingen bij uw PC. Basiskennis is een minimaal vereiste om niet onder te sneeuwen in het D.C.-tijdperk. In technische termen, om niet afhankelijk te worden van andermans netwerk software, maar eigen D.C.-routines vanuit uw eigen programmatuur te kunnen aanroepen.

Nieuwe chips

Nieuwe chips maken het u daarbij gemakkelijker. Zoals de 8052 van Intel: een chip op 12 MHz, met als 'native code': Basic... Dus gemakkelijk programmeerbaar om I/0 (D.C.) routines af te handelen. Kortom voor circa fl. 50,- (naar verwachting) per chip haalt u evenzovele 'paralelle processoren' 'aan boord' - precies in lijn met de verwachte ontwikkeling van de computer: parallelle processing. (Zie latere artikelen in Aarde en Kosmos.)



Telefoontoestellen; op weg naar een steeds moderner uiterlijk en vele verschillende toepassingen die alles met data-overdracht te maken hebben.

Volgend artikel:

de hardware (chip structuur).

Telexverkeer via de computer

De leverancier van automatiseringssystemen ORMAS te Houten ontwikkelde voor haar Sharp OA en Fortune Unix-computersystemen het softwarepakket ORTELEX. Dit programma, dat gebaseerd op Ormas'tekstpakket ORTEXT, stelt gebruikers van de genoemde systemen in staat uitgaande telexberichten via de computer aan te maken en te verzenden.
Tevens kunnen inkomende telexberichten via de computer worden bekeken en desgewenst worden uitgeprint of weggeschreven worden als een ORTEXT-document.
Daarnaast kunnen ook standaard ASCII files onder Unix rechtstreeks op het telexnet worden gezet. Het systeem houdt een logboek bij waarin alle inkomende en uitgaande telexberichten worden geregistreerd, gesorteerd op naam, volgnummer, telexnummer en datum en tijd. Erg handig dus.

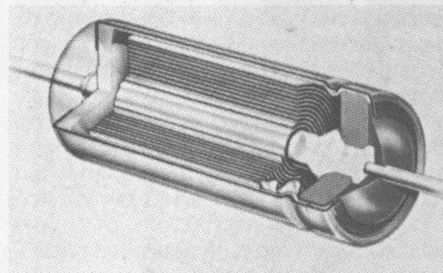
Nieuw draadloos modem

Wie geen zin heeft, iedere keer als hij data wil overzenden vanuit zijn draagbare PC, naar een telefooncel te gaan om dit te volbrengen, kan binnenkort gebruik gaan maken van een splinternieuw modem (genaamd 'Estreem') dat de firma Impact te Utrecht gaat leveren.
Men hoeft geen kabelverbinding meer te gebruiken om met een ander systeem te communiceren. Daar waar de kabelstructuur geheel ontbreekt, is dit een praktische oplossing. Te denken valt bijvoorbeeld aan toepassingen in de off-shore, mobiele hijskranen en vliegtuigen. De mogelijkheden van een mobiel modem worden zelfs al in de autoracesport (Formule-1) toegepast.
U hoeft natuurlijk niet in het bezit te zijn van een Formule-1 bolide om toch van deze mogelijkheid gebruik te maken want een bijgeleverde mobiele antenne, voorzien van een magneetvoet, kan ervoor zorgen dat u ook vanuit uw wagen kunt communiceren met bijvoorbeeld uw kantoor of gewoon met thuis. Het modem kan probleemloos full-duplex werken en is instelbaar in 7 baudreeksen, te weten 110, 300, 600, 1200, 2400, 4800 en 9600 baud.
De zend- en ontvangfrequentie is instelbaar tussen de 72.020 en de 72.980 MHz en dit alles frequentie gemoduleerd.



Droge elco's mogen de ruimte in

De droge elektrolytische condensatoren van de 123-reeks zijn officieel goedgekeurd door de European Space Agency (ESA).
Daarmee is de 123-reeks op de lijst van gekwalifiveerde produkten van de Europese ruimtevaartorganisatie gekomen. Alleen produkten die op deze lijst staan mogen in ruimtevaartuigen worden gebruikt. Omdat deze kwalificatie niet lichtvaardig wordt afgegeven, betekent opneming op deze lijst dat deze serie droge elektrolytische condensatoren aan de hoogste eisen ten aanzien van kwaliteit en bedrijfszekerheid voldoen.
De aluminiumcondensatoren hebben axiale aansluitdraden. Dankzij hun hoge bedrijfszekerheid zijn ze uitverkoren voor gebruik in de ruimteraket Ariane. Bedrijfszekerheid was niet het enige punt waarop de 123-reeks hoog scoorde. De condensatoren hebben ook stabiele eigenschappen en een lage drift; dat is gebleken bij levensduurtests, waarbij de condensatoren gedurende 5000 uur (meer dan 100 dagen) bij een temperatuur van 125 graden Celsius zijn getest. Ook een 2000 uur durende test bij 150 graden Celsius behoorde tot de beproevingen. Dit betekent in de praktijk dat temperaturen tot 175 graden Celsius toelaatbaar zijn.
Bovendien zijn deze condensatoren goed bestand tegen grote rimpelstromen en spanningen in tegenrichting en hebben ze een lange levensduur.



Supermagneet voor medische research

In het medisch research-centrum van Siemens te Erlangen (West-Duitsland) is een supergeleidende magneet ontwikkeld (sterkte 4 Tesla).
De magneet zal worden toegepast bij kernspintomografie voor het hele lichaam. Er is reeds een begin gemaakt met de eerste proeven.
De 4 Teslamagneet, die dubbel zo krachtig is als de tot nu toe sterkste kernspintomografiemagneten, opent voor het medisch onderzoek nieuwe mogelijkheden op het gebied van de beeldvormende diagnose.
Terwijl tot dusverre voor het produceren van kernpintomologie-beelden gebruik werd gemaakt van de eigenschappen van de in het lichaam rijkelijk aanwezige waterstof, is het met magneten met krachtige velden mogelijk de kernen van andere, minder vaak voorkomende elementen in het lichaam aan te spreken zoals bijvoorbeeld koolstof, fluor, fosfor of natrium die als specifieke indicatoren voor bepaalde stofwisselingsprocessen dienen. Met de nieuwe magneten kunnen dergelijke processen in het menselijk lichaam zichtbaar worden gemaakt.
De supergeleidende magneet is 2.80 meter lang en heeft een opening van 1.25 meter.

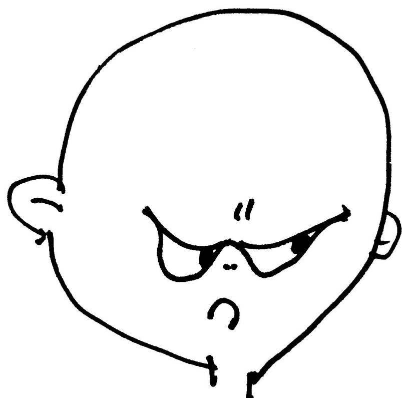


D. Vos
Siso code 365

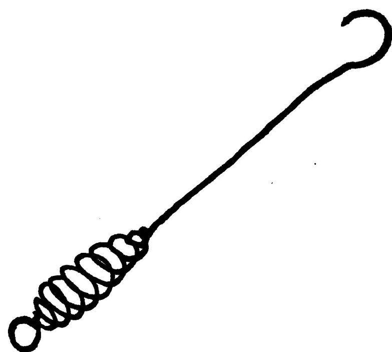
BASIC CURSUS

DEEL II - 5

In deze les wordt aangegeven hoe u het Basic-programma zelf kunt verkennen (de geheugenlocatie nalopen). En hoe de geheugenlocaties in waarde kunt veranderen. Een van de toepassingen van dit laatste is het maken van een eenvoudige 'vierde generatie'-taal, of uw eigen taal naar keuze.



PEEK betekent gluren, of kijken op geheugenlocaties.



'POKE' betekent 'porren' of het instellen van geheugenwaarden.



GO FORTH betekent niet 'ga weg' maar 'stap over op FORTH'.

PEEKER

Voor het openleggen van het geheugen schrijven we eerst een 'peeker'-routine. Deze routine 'kijkt' op opeenvolgende geheugenlocaties en print de daar gevonden waarden achterelkaar uit. Voorbeeld:

```
50 FOR I=4150 TO 4200
60 PRINT PEEK(I);" ";
65 IF PEEK(I)=0 THEN PRINT
70 NEXT I
```

Opmerkingen:
① de beginwaarde I=4150 geldt slechts voor de IBM PC (Basica onder MSDOS) - voor andere machines en Basics zijn andere beginwaarden beter geschikt. Toch is het leuk om alle geheugenwaarden eens te doorlopen. U bent nu ook in staat om routines te schrijven om op al die geheugenlocaties waarden in te 'poken' (tussen 0 en 255) en te kijken wat er gebeurt. Zo iets is leerzaam en er kan niets verkeerd gaan: als het programma vastloopt, start u MSDOS gewoon opnieuw op.
② De locaties 4150 tot 4200 gelden vooral voor het onderstaande geval: een stukje Basic komende vóór regel 50, waarvan de inhoud straks door pokes wordt veranderd.
③ Regel 60 print de geheugenwaarde uit met een spatie ertussen.
④ Als die waarde nul is, volgt een nieuwe regel door een extra printopdracht.

GOSUB (variabele)

De bedoeling is via poking een routine tot stand te brengen: GOSUB (variabele). Met zo'n routine kan een hogere taal gebouwd worden (hoe langzaam ook). Een tekst file kan bijvoorbeeld nummers bevatten, die nummers worden ingelezen, en als variabele ingepoked op de plaats GOSUB (variabele). Het wordt dan een soort 'FORTH'-achtig taaltje, of, wat dat betreft, een taaltje precies zoals u wilt. U kunt LOGO simuleren of elke andere taal, ervan afhankelijk hoeveel werk u wilt steken in het leesmechanisme (dat de tekstfile 'leest'), en of u extra variabelen vanuit de tekstfile wilt doorgeven naar de GOSUB routines. Er bestaat ook een statement:

```
On X GOTO y1, y2, y3 enz.
```

Deze is echter hiervoor niet of veel minder geschikt. Bij elke nieuwe routine moet deze regel zelf aangepast worden, bovendien moet x opeenvolgend genummerd worden (1,2,3 enz.) en kan de routine niet rechtstreeks regelnummers meekrijgen, zoals in GOSUB (variabel regelnummer). Nog beter dan regelnummers zou het zijn om routinenamen te gebruiken, wat in nieuwere basics probleemloos kan.

BRON CODE

Het stukje bron code (oorspronkelijk programma in tekst format) dat aangepast gaat worden is als volgt:

```
20 IF S=0 THEN GOTO 1300: 'voor opstarten
30 GOSUB 4000: 'wordt veranderd door pokes
32 RETURN
40 TO=4193: POKE t0+0,14: POKE t0+1, X2:
POKE t0+2,X1:
42 RETURN
```

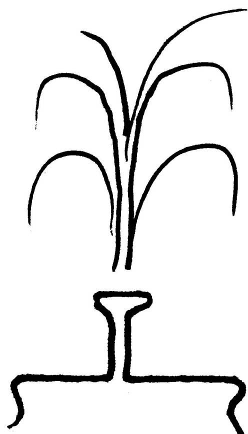
Opmerkingen:
① Zoals bekend dient regel 20 alleen voor de eerste keer dat het programma wordt opgestart.
② De regel 30, is de regel waarvan de waarde steeds veranderd gaat worden. In plaats van 4000 staat dus steeds een andere waarde 'onder' het getal 4000. Elke keer dat 'list' wordt gegeven is echter alleen '4000' te zien. (Om deze reden wordt in de volgende COBOL standaard een overeenkomstige statement ('ALTER') uit de standaard verwijderd: het onderhoud wordt hierdoor moeilijk omdat de listing niet echt te zien geeft wat er in het programma staat. Andere talen bieden deze mogelijkheid gelukkig nog wel, en het is altijd mogelijk om op machine-taal routines terug te vallen - bij gecompileerde versies is dit zelfs noodzakelijk.)
③ Regel 40 'poke-t' de nieuwe regelnummerwaarde in op/vanaf lokatie 4193.
④ Let op dat wijziging van bijvoorbeeld regel 20 de waarde 4193 weer meeverandert!
⑤ Op lokatie 4193 (TO + 0) wordt byte met waarde 14 ingepoked. Een andere waarde is mogelijk (er zijn twee keuzen) ervan afhankelijk of het programma reeds 1 maal gerund is, of niet. Na de eerste run wordt de waarde CHR$(14), die in het bovenstaande voorbeeld, dus maar meteen wordt ingepoked.
⑥ Op de volgende twee locaties wordt het regelnummer ingegeven, en wel in twee bytes. De eerste byte telt gewoon, de tweede byte x 256. Dit is precies anders-

om als bij een gewoon talstelsel, waar de hogere tallen eerst komen. De waarde van dit getal komt uit een tekstfile, waar het nummer 'gewoon' (decimaal) staat. Het wordt daar gesplitst in een veelvoud van 256 (x1) en de rest (x2). Vervolgens wordt op regel 40 eerst x2 ingepoked en dan x1.

Proberen van de peeker

De peeker kan geprobeerd worden met:
10 GOTO 50

Geeft u nu RUN, dan verschijnen series getallen op het scherm: dit zijn de ASCII waarden van de bytes die het Basic programma vormen. Probeert u eens de relatie te leggen met de Basic listing zoals u die gewoonlijk kent.



'BRON CODE' is het programma als gewone tekst file, niet in machinetaal (0/1).

Opmerkingen:
verwijder regel 10 weer, als u de waarde 4193 ongewijzigd wilt laten.

Opstarten van het programma

Bij het opstarten van het programma moeten eerst de voor de menu's benodigde strings worden gedefinieerd. Dan kan het programma in een keer naar een met regelnummers gevulde tekstfile lopen, die gaan lezen, en de desbetreffende routines achterelkaar aanroepen. Nog mooier is het de gebruiker de gelegenheid te geven eerst een tekstfile met regelnummers te vullen en de subroutines op die regelnummers te gaan doorlopen. De gebruiker moet er wel op letten dat die regelnummers ook echt bestaan (of moet 'woorden' gebruiken die door het programma zelf in bestaande regelnummers worden omgezet - niet-bestaande regelnummers of -'woorden', worden eenvoudig overgeslagen).De gebruiker kan ook nieuwe subroutines naar behoefte creëren: hij heeft nu een 'uitbreidbare' taal.
De 'grammatica' van de taal kan hij ook zelf bijstellen, mits hij maar de leesroutine kan aanpassen (als 'parser', een routine dat de tekstfile verdeelt in stukjes informatie die bij elkaar horen). Door een stukje tekstfile met één woord aan te duiden, kunnen woorden in een tekstfile ook verwijzen naar andere tekstfiles, ofwel 'functieblokken'.
Uit de flexibele industriële automatisering blijkt, dat het gebruik van functieblokken het beste werkt, het gemakkelijkst voor de gebruiker te leren is, en het snelst en goedkoopst is.

Parameters

De voor de functieblokken benodigde parameters worden uit een bepaalde geheugenlocatie gehaald en de resultaten van de functieblokken weer naar een bepaalde geheugenlocatie weggezet. In de bovenstaande tekstfilefunctieblokken kan dit eveneens, waarbij in de tekstfile de Basic geheugenlocaties zelf staan, of de stringnummers waar de parameters thuishoren. Voorlopig wordt alleen een beschrijving gegeven van functieblokken zonder parameters.

Snelheid

In Basic is deze structuur niet geschikt voor snelle toepassingen, maar dat is Basic toch al niet.
De structuur van het geheel is echter herhaalbaar in andere talen, zoals C en Pascal, maar ook in machinetaal (machinetaal is op dezelfde wijze gestructureerd). Hierdoor kan èn flexibiliteit èn snelheid worden verkregen, zodat het systeem ook geschikt wordt voor redelijk kritisch 'real time' toepassingen. Zijn er hardware interrupts (bij aansturen van een bepaalde lijn, springt het programma automatisch naar een bepaalde routine) dan is men helemaal ingedekt.

Voorbeeld in Basic

Bovendien zijn veel toepassingen helemaal niet kritisch, vanaf 'Slow scan TV' tot het langzaam omhalen van schakelaars in een huis, of één keer per uur bepaalde schakelaarstanden checken (dat allles wordt met dit systeem mogelijk door eenvoudig de desbetreffende Basic routines te schrijven). Laten we eens zien hoe een voorbeeld in basic eruit zou zien. In het onderhavige voorbeeld zijn de routines hiervoor op de volgende regelnummers gezet:

12200 REM (')
begin van het leesgedeelte
12215 IF S=0 THEN S=1:
S(1)=501: s(2)=1:
GOSUB 12300

Opmerkingen:
① het stukje over s=0 heeft betrekking op gebruik van deze routine als het programma opgestart wordt; het kan anders worden weggelaten (als men de routine vaker wil gegbruiken, dus ook als de gebruiker zijn eigen tekstfile ingevuld heeft).
② S(1)=501 geeft de (string-)locatie aan waar de rest van de leesroutine de te lezen tekstfile verwacht. (U kunt ook andere vrije ('in-array') locaties invullen, maar uw eigen tekstfile ook op locatie 501 laten beginnen, naar keuze.)

Woorden inlezen

Er volgt een routine op 12300 die de ingelezen woorden (nu nog regelnummers) interpreteert en naar de desbetreffende subroutines springt, maar deze begint met een routine om de woorden vanuit de tekstfile in te lezen, op regel 11190.

11190 X1=1: Y1=X1: X2=s(1)+s(2)+1: X3=2
11194 GOSUB 970
11195 IF Y1=0 THEN Y1=LEN(s$(X2))+1
11196 IF Y1>X1+1 OR Y1=0 THEN
S$(Y2)=MID$(S$(Y2), X1, Y1-X1): RETURN
11198 X1=Y1+1: GOTO 11194

en de routine op 970:
970 Y1=INSTR(X1, S$(X2), S$(X3)): RETURN

De bijbehorende opmerkingen zijn als volgt:
① Op regel 11190 worden als X1, X2, en X3 de parameters gedefinieerd die de routine op 970 nodig heeft, het resultaat van die routine komt weer terug als Y1. (Deze methodiek loopt vooruit op de behoefte om de parameters die vanuit een tekstfile aan routines moeten worden doorgegeven, één-éénduidig aan te geven.)
② De routine op 970 geeft (als Y1) retour, de plaats (y1) in de string S$(X2) waar S$(X3) voorkomt (de laatste string is "Gosub"). Na het woord GOSUB behoort in de tekstfile het regelnummer gezet te zijn, waar de betreffende subroutine zich behoort te bevinden. Weten we dus de plaats waar het woord GOSUB staat, dan kunnen we uitrekenen waar het regelnummer staat, dat stringgedeelte inlezen, converteren naar een getal, en het getal doorgeven aan de routine op regel 40 (dus springen naar het ingelezen regelnummer).
③ Regel 11195 maakt als het woord GOSUB niet in de regel voorkomt (y1=0) y1 groter dan de regellengte (waardoor straks een lege string wordt aangemaakt).
④ regel 11196 maakt een string S$(Y2) waarin het "Gosub" zit. Deze string S$(Y2) wordt nu verder verwerkt in routine 12300.

Het 'runnen' van een tekstfile

Routine 12300 'runt' als het ware een tekstfile met gosub-regelnummers.

```
12300 REM run een tekst file
12310 GOSUB 11190: 'lees S$(y2)
12318 IF S$(Y2) "End" THEN GOTO 12395
12320 IF S$(Y2)<>"GOSUB"THEN GOTO 12310
12322 IF Y1=0 THEN GOTO 12390
12330 X1=Y1: GOSUB 12294
12340 Y3=VAL(S$(Y2))
12350 IF Y3=0 THEN GOTO 12390
12360 Y4=INT(Y3/256): Y5=Y3-Y4 * 256
12370 X1=Y4: X2=Y5
12380 GOSUB 40: GOSUB 30
12390 S(2)=s(2)+1: A=FRE(0): GOTO 12300
12395 RETURN
```

Tekst en uitleg volgen:
① Regel 12310 leest de string S$(Y2) in.
② Als de string gelijk is aan "End", wordt de tekstfile niet verder gelezen (12318).
③ Als de string ongelijk is aan "Gosub", wordt het volgende woord ingelezen (er kunnen dus andere woorden ingevoegd zijn die niet worden 'geïnterpreteerd').
④ Als er geen Gosub in de regel voorkwam (Y1=0), dan wordt via regel 12322 en 12390 de volgende regel gelezen. Het A=FRE(0) wordt alleen tussengevoegd om regelmatig stringclearing te doen, zodat de executie niet opeens wat langdurig

onderbroken wordt voor alle achterstallige string huishouding.
⑤ Let op dat End en Gosub precies zo geschreven moeten worden. END en GOSUB tellen dus (nu nog) niet. Dit kunt u zelf verhelpen met OR op de desbetreffende plaats, of door liefst een routine te gebruiken die de undercast omzet naar uppercast en vergelijkt in uppercast.

Het inlezen van het regelnummer

Op regel 12330 X1=Y1: GOSUB 12294 wordt gesprongen naar een routine 12294 die het volgende woord inleest en retourneert als een getal (regelnummer waar straks heen gesprongen gaat worden).
Deze routine is geheel gelijk aan de routine op regelnummer 11194 (feitelijk zou deze opnieuw gebruikt kunnen worden). Het resultaat is S$(Y2), met dit keer in S$(Y2) een getal. (Dit is het regelnummer waarnaar gesprongen gaat worden.)

Verder met 12300

Op regel 12340 wordt de waarde van dit getal toegekend aan Y3. Als de waarde nul is wordt de subroutine via 12350 en 12390 verlaten.
Verder mag worden aangenomen dat de waarde niet nul is. In de regels 12360 t/m 12370 worden de waarden X1 en X2 gevormd. X1 bevat het aantal malen dat 256 in het regelnummer zit, en X2 de rest. Via regel 12380 worden deze parameters doorgegeven aan regel 30 en 40, op de besproken wijze. Dat wil zeggen, eerst worden de parameters doorgegeven en vervolgens worden naar regel 40 gesprongen. Daar staat GOSUB 4000, wat nu is veranderd in GOSUB Y3.

De toepassingen

Tot dusverre kan iemand zich nog afvragen wat het nut is. Dan heeft hij nog niet echt nagedacht. De eindgebruiker kan nu in een tekst file een programma schrijven en uitvoeren. Hele programmastukken kunnen op een regelnummer gezet worden, en daar aangeroepen.
Tekst files (met stukjes programma) kunnen in een database worden gezet en naar believen (zelfs tijdens het draaien van een programma) worden geladen en 'gerund'. Kortom in principe hebben we een '4G' taal gemaakt (zelfs een die naar keuze te definiëren en uit te breiden is).

Nogmaals parameters

Het is nog wel een simpele taal, alleen bestaande uit reeds opgebouwde routines. Voorbeeld:

Gosub 1000
Gosub 2000
Gosub 3000
End.

De taal kan worden uitgebreid met een mogelijkheid om parameters door te geven, bijvoorbeeld Gosub 25000 Param 20
In het leesgedeelte moet dan een manier komen om die parameters op een standaard manier door te geven aan de gekozen subroutine. Dit kan door die subroutines de parameters te laten opvragen X1, X2, enz. Of nog liever X1$, X2$, X3$ en in de subroutine te laten converteren naar de numerieke waarde (getalswaarde), indien van toepassing. De vraag resteert hoeveel parameters per routine moeten worden doorgegeven, en wat er gebeurt als een parameter niet wordt gedefinieerd (of één te weinig). De waarde is dan nul, of een andere, oude waarde.
Het is natuurlijk ook mogelijk de parameters één voor één te definiëren in evenzovele subroutines. In het geval van meervoudige parameters ziet de taal eruit als de manier waarop een besturingssysteem

Schrijf Jan
Schrijf Jansen
Uitwissen

In principe hebben we de elementen voor een eigen, flexibele en uitbreidbare taal met alle voordelen (en nadelen) vandien.

wordt aangestuurd (zoals MSDOS) en de manier waarop C is opgebouwd, in het tweede geval lijkt het meer op Forth (de parameters moeten worden gedefinieerd, vóórdat de subroutine wordt aangeroepen die al die parameters gebruikt) of zelfs op een Job Control Language van een mini (soort batchtaal, voor achtereenvolgende bewerkingen). Het verschil met de laatste, is dat de tekstfile natuurlijk ook tijdens de executie gewijzigd worden kan! (Dit kan het resultaat zijn van een bepaalde subroutine)Het bevestigt in elk geval het idee dat batch-programma's tegenwoordig niet minder zijn dan de laagste prioriteit job die op de computer draait.

Ontwikkeling contra 'produktie'

Het bovenstaande heeft dus veel meer met taalontwikkeling te maken (van een computertaal), dan met 'produktie' (het bedrijfsmatig gebruik van programma's geheel los van de ontwikkeling). Voor produktie is nodig programmatuur waaraan geen byte meer te wijzigen is. Ook dit kan worden bereikt door de latere correctieoptie af te splitsen van de input. (Het moet niet vanzelfsprekend zijn dat iemand met inputbevoegdheid ook kan muteren. Boekhouders geloven hier in elk geval zeker niet in, en ze hebben gelijk.)

Gebruik

Als u de beschreven mogelijkheden gaat gebruiken, neem dan de subroutines zoals die in deel I van deze Basiccursus zijn gedefinieerd. U zult wel zelf een routine moeten schrijven om de parameters door te geven.
Iets anders is hoe deze parameters kunnen worden doorgegeven als u uw Basicversie wilt gaan compileren. In de volgende aflevering II-6 e.v. wordt ingegaan op dit probleem: de vele diverse soorten Basic, hun eigenschappen en met name of parameters aan de objectcode kunnen worden doorgegeven. Daarna komt een simpele conversie-problematiek aan de orde, afhankelijk van de gebruikte taal, en het gebruik van het Operating Systeem. □

Return Info

Hebt u vragen?
Suggesties?
Ideeën?
Aarzel niet, maar stuur ze naar de redactie van
A&K-INFORMATICA
Postbus 108 - 1270 AK Huizen

Rien van Dongen

Gebedstijden

Gerard van Rossum uit Paterswolde meldt dat het programma Islamitische Gebedstijden op zijn Amstrad Schneider 6128 computer niet loopt, omdat er de Basic-opdracht 'DI' in voorkomt. DI wordt gebruikt als dag van de Arabische datum. Het commando schakelt bij deze computer een interrupt uit en dan ontstaan er problemen. De heer Van Rossum stelt voor om RDI als variabele te gebruiken in de regels 2720 t/m 4160. Verder merkt hij op dat zijn computer weinig problemen heeft met de ontvangst van zondagavonduitzendingen van Basicode-2 programmatuur van Hobbyscoop. (Als de Optimod tenminste uitstaat.)
De veranderingen betreffen de volgende regels:

```
2720 REM arabische datum (RDI,MI,JI)

2840 RDI=L4-INT(29.54*MI-29)+F
2850 IF RDI=0 THEN RDI=30:MI=MI-1
2860 IF MI=0 THEN MI=12:RDI=30:JI=JI-1

3140 CT=2:CN=0:SR=RDI:GOSUB 310

3240 CT=2:CN=0:SR=RDI:GOSUB 310

4090 CT=2:CN=0:SR=RDI:GOSUB 310

4160 CT=2:CN=0:SR=RDI:GOSUB 310
```

In Basicode mogen de namen van variabelen maximaal twee tekens lang zijn (tekstvariabelen krijgen daarachter nog een dollarteken). Wanneer een bepaalde verandering alleen geldt voor een bepaald type computer, mag een drie-letternaam als RDI natuurlijk best, als die computer dat begrijpt.

Op het ogenblik zijn in Basicode de volgende namen voor variabelen verboden: AS, AT, FN, GR, IF, LN, PI, ST, TI, TI$ en TO. Vooral het gebruik van ST is verleidelijk, wanneer het om een stapgrootte gaat. Maar niet doen dus! Eigenlijk zou nu DI aan de lijst moeten worden toegevoegd. Wanneer de Amstrad Schneider 6128 had bestaan toen Basicode ontstond, zou DI ook verboden zijn geweest. De makers van Basicode konden echter niet voorzien dat die lettercombinatie ooit nog ergens een speciale betekenis zou krijgen.
(H.G. Janssen) □

Vier MegaHertz Power Board

Voor nog geen vijfhonderd gulden is er een nieuwe opvoerset voor de C-64 te verkrij-

gen. In tegenstelling tot de meeste andere turbo's voert deze set niet alleen het tempo van de disk-drive flink op maar laat hij ook de computer zelf viermaal zo snel werken. Het werk van de processor wordt namelijk overgenomen door een andere, 16-bits processor. Daardoor is nu ook 16 Megabyte aan geheugen adresseerbaar. Op de kaart die gewoon achter in de uitbreidingspoort geschoven kan worden, zit 64-k extra RAM-geheugen dat door middel van batterijtjes onder spanning blijft staan, ook na het uitzetten van de computer. Later kan gemakkelijk 256-k RAM bijgeplaatst worden. Spelletjes zijn bij het zeer hoge tempo soms niet meer te volgen. Daarom is ook het regelen van de snelheid mogelijk; van een tiende tot driemaal de normale snelheid.
Inlichtingen: Cat & Korsh, 010-4765870 □

NaBoNT, nascholing beroepsonderwijs

Opleidingsinstituten kunnen de veranderingen in het bedrijfsleven niet meer bijhouden. De kennis van de docenten is al verouderd op het moment dat ze hun leraarsdiploma in ontvangst nemen. Het bedrijfsleven klaagt dat de opleidingen niet goed aansluiten bij de huidige technologie. Daarom is NaBoNT in het leven geroepen. Uitgaande van de vraag van het bedrijfsleven worden cursussen gegeven. De cursisten bepalen zelf of de cursussen in hun behoeften voorzien. Een totaal andere benadering van onderwijs. Toch staat het reguliere onderwijs niet buiten spel. Samen met NaBoNT kunnen cursussen en projecten worden georganiseerd. Zin in samenwerking met ministeries en bedrijfsleven? Behoefte aan het volgen of geven van cursussen?
NaBoNT, Koninginnegracht 1, 2514 AA Den Haag. □

Schneider gebruikt DI-variabele

Schneider computers hebben moeite met BASIC-programma's waarin de variabele DI voorkomt. Die wordt al gebruikt voor de actuele datum. Iemand die met een andere computer BASICODE-programma's maakt, moet DI dus niet gebruiken. Het is toch al niet zo handig om de letter I in variabele namen te gebruiken omdat die erg veel op een L of een 1 lijkt. □

Pauze tussen pieptonen

Soms worden twee BASICODE-programma's met een te korte pauze uitgezonden. De pieptoon van het eind van het eerste programma zit dan te dicht op die van het begin van het tweede programma. Bij het laden kan dat problemen opleveren. We vragen Hobbyscoop om de programma's altijd door een duidelijke pauze van elkaar te scheiden. □

Keynes' economie, niet te vergelijken

Bijna tegelijkertijd met het artikel 'Economie, een kwestie van vergelijken', werd door de TROS een prachtig computerprogramma over de Keynesiaanse economische theorie uitgezonden. Dit programma zal zeker op een verzamelcassette van BASICODE-3 programma's terecht komen. Maar het is zeker de moeite waard het nu al te bemachtigen. En de minister maar denken dat er geen goede onderwijssoftware bestaat! □

BASICODE vertaalprogramma voor de C-16!

Binnen een maand na de oproep in A&K Informatica was er een programma waarmee BASICODE programma's in de Commodore C-16 geladen kunnen worden. Robert Mast kon niet langer aanzien dat duizenden C-16 gebruikers het zonder BASICODE zouden moeten stellen. Zijn programma kan wel BASICODE lezen maar nog niet schrijven. Daar werkt hij ongetwijfeld hard aan. In een standaard C-16 zit maar 16-k geheugen en daarin kan naast het vertaalprogramma een 12-k programma geladen worden. Met een geheugenuitbreiding of op de Plus 4, het broertje van de C-16, kan bijna 60-k geladen worden. Gefeliciteerd Robert, een mooie prestatie!
Inlichting: Robert Mast, Klagerstuin 76, 1689 JS Zwaag. □

INPUT en de gebruiker

In sommige programma's moet de ene keer de invoer van het toetsenbord mèt de INVOER-toets (RETURN- of ENTERtoets) afgesloten worden en de andere keer niet. Een enkele letter, bijvoorbeeld bij een menukeuze, wordt niet, een getal of woord wèl afgesloten met die toets. Dit leidt tot verwarring bij de gebruiker. Het resultaat is meestal dat er te veel op de INVOERtoets wordt gedrukt. In die programma's wordt meestal de inputbuffer (een stukje geheugen waarin de toetsaanslagen tijdelijk worden bewaard) niet geleegd en blijft de extra toetsaanslag bewaard. Wordt hierna weer om invoer gevraagd, dan wordt de opgeslagen toetsaanslag meteen verwerkt hetgeen tot rare resultaten of fouten aanleiding kan geven. Een vaak voorkomend effect is het te snel voorbij flitsen van pagina's omdat de INVOERtoets ook wel gebruikt wordt om aan te geven dat een pagina tekst gelezen is (een soort VERDER-toets). Laat daarom invoer van het toetsenbord altijd afsluiten met de INVOERtest of maak steeds de inputbuffer leeg. In BASICODE-programma's gaat dat heel eenvoudig: FOR BF=1 TO 10;gosub 200; NEXT BF. □

BASICODE-3 voor MS-DOS en MASTER

Het heeft even geduurd maar het aantal vertaalprogramma's voor BASICODE-3 begint zich uit te breiden. Er is nu BASICODE-3 voor de BBC-master en zeer binnenkort komt er een echt vertaalprogramma voor de MS-DOS PC's. □

ASCII-files op Commodore Disk

Een BASIC-programma wordt in het computergheugen anders opgeslagen dan we het op het scherm zien. Alle BASIC-woorden worden afgekort tot één teken waarvan de getalwaarde ergens boven die van de standaard ASCII-tekens ligt. Vooral omdat de gebruikte tekens per computertype verschillen, is dat slechts handig zolang de programma's voor BASIC-verwerking in die ene computer dienen. GW-BASIC-programma's kunnen op MS-DOS computers gemakkelijk als ASCII-file (zoals we ze op het scherm zien) op schijf gezet worden: SAVE'programmanaam',A waarbij de A voor ASCII staat. Daarna kunnen ze als een gewoon tekstbestand verwerkt of overgeseind worden. Ook bij Commodore computers is dat niet echt moeilijk: OPEN 3,8,3,'programmanaam,S,W':CMD 3:LIST 1000-12000 en als de cursor weer knippert de operatie afsluiten met PRINT3;CLOSE3. Doordat het LIST commando volledig beschikbaar is kan ook een deel van het programma omgezet worden in ASCII. □

Aardrijkskunde leraren oneens

De heer W.F. Jansen uit Breda is het niet eens met de heer Heinecke uit Waalwijk (A&K Informatica 3-1987). Hij vindt het juist heel interessant om als ex-aardrijkskundeleraar de programma's op zijn kleine MSX computer te gebruiken. Wel vindt ook hij de andere artikelen in het totaalblad Aarde en Kosmos zo interessant dat hij liever dat combinatietijdschrift leest dan alleen maar het informatica deel. □

ATARI-600 verwaarloosd?

J. Jacobs uit Klimmen ziet liever wat meer programma's voor de ATARI-600. De mooiste en leukste programma's uit A&K Informatica kan hij niet gebruiken omdat er voor deze computer geen BASICODE bestaat. Het is zeker niet zo dat we geen programma's voor de genoemde computer willen plaatsen. Maar de programma's worden allemaal speciaal voor dit blad ontworpen en geschreven. Het zou niet slim zijn om programma's te maken voor een betrekkelijk kleine groep gebruikers en wel zo dat de anderen ze dan weer niet kunnen gebruiken. BASICODE werkt op meer dan 90% van de computers van onze lezers. Het moet echter in de meeste gevallen heel goed mogelijk zijn om de programma's in te tikken en ze werkend te krijgen. De ATARI-600 kan in principe alles wat nodig is voor onze programma's. Mocht de gebruikersgroep u niet kunnen helpen, dan willen wij wel eens naar de overeenkomsten met uw ATARI-BASIC kijken. Daarvoor hebben we dan wel een beschrijving van alle ATARI-BASIC opdrachten nodig. Probeer het echter eerst eens bij de gebruikersgroepen:
HCC Atari gebruikersgroep, Raadhuislaan 114, 2251 GE Voorschoten.
Atari gebruikers stichting, Postbus 40181, 6504 AD Nijmegen. □

Computer voor de studie

Martijn Hendriks uit Landgraaf wil later aan een technische universiteit gaan studeren. Twee jaar voor zijn eindexamen wil hij een computer kopen die hij ook vóór zijn studie kan gebruiken. We geven geen koopadvies maar wel een algemeen antwoord op deze vraag. Op een technische universiteit zul je zeker computers gaan gebruiken. Daarvoor heeft de universiteit zalen vol met apparaten staan. Voor de ingewikkelder zaken zijn er zelfs heel krachtige computers aanwezig. Daarvoor hoef je dus geen computer te kopen. Als je een studierichting kiest waarbij een computer thuis nodig is, dan gaat het

daarbij meestal om een bepaald type dat je door bemiddeling van de universiteit met korting kunt kopen. Er valt nu nog niets te zeggen over het type dat over enige jaren wordt aanbevolen. De techniek vordert zo snel dat de PC van nu over een jaar of drie volslagen verouderd is.
Maar waarschijnlijk wilde je een ander antwoord omdat je zo snel mogelijk met de computer aan de gang wilt. Welnu, neem dan een goedkope maar universele computer waarmee je allerlei zaken die op een technische universiteit aan de orde komen op kleine schaal kunt uitproberen. Daarbij zijn de volgende overwegingen van belang. Een computer waar er veel van verkocht zijn, heeft het voordeel dat je met veel mensen kunt samenwerken en dat er zeer veel goedkope software verkrijgbaar is. Neem in ieder geval een computer die BASICODE-2 en -3 kan gebruiken. Via de radio krijg je heel vaak technische programma's gratis aangeboden. Waarschijnlijk krijg je tijdens je studie te maken met tekstverwerking, databases en spreadsheetprogramma's, maar ook met het ontwerpen van schakelingen of constructies (CAD). Er zijn niet zoveel kleine computers die daar aardige en goedkope programma's voor hebben. Er blijven nog minder typen over als we zoeken naar een computer die behalve schakelingen (voor joysticks) ook analoge ingangen standaard heeft ingebouwd. Daarmee kun je bijvoorbeeld spanningen meten of zelfs een osilloskoop van je computer maken. Communicatie via de telefoon is erg in en heel leerzaam. Er moet dus ook een betaalbaar modem (tussenschakeling voor computer en telefoonnet) en een goed communicatieprogramma beschikbaar zijn. Ten slotte is de aansluiting voor een printer, een plotter (grafiektekenmachine) en een uitgang om een robotje te besturen heel erg handig. Geef in ieder geval nu niet veel geld uit in de hoop dat je dezelfde computer straks bij de studie kunt gebruiken! Je studieboeken koop je toch ook pas over twee jaar? □

BASICODE-3 voor de Schneider CPC 464

Er is een BASICODE-3 vertaalprogramma verschenen voor de Schneider CPC 464 computer. Bezitters van deze computer kunnen het in het bezit krijgen door f. 15,- over te maken naar de maker van het programma, Jef Simons uit Etten Leur. Zijn gironummer is 2969984. Vermeld BASICODE-3 Schneider CPC.
Voor de meeste andere computers (onder andere Commodore 64, Spectrum, MSX, P2000, BBC) ligt de sleutel tot BASICODE-3 in het boek BASICODE-3, dat compleet met een cassette in de boekhandel te koop is voor f. 27,50 (uitgave Kluwer Technische Boeken, ISBN 90-2-1-1949-4). □

Software

Het officiële BASICODE-3 vertaalprogramma voor IBM-PC/MS-DOS computers is eindelijk uit! Het is gemaakt door Hans Vosman en kost f. 15,-. U kunt dat bedrag overmaken op bankrekening 32.82.67.783 ten name van ABC Software Service te Hilversum, onder vermelding van BASICODE-3 IBM PC. Houders van een girorekening f. 15,- overmaken op giro 5548948 ten name van J. Puipe te Hilversum, weer onder vermelding van BASICODE-3 IBM PC.
Zolang u het officiële vertaalprogramma nog niet in huis hebt, kunt u proberen BASICODE-programma's uit dit tijdschrift aan het draaien te krijgen door voorafgaand aan regel 1000 deze GWBASIC opdrachten te plaatsen.

```
10 SCREEN 0:WIDTH 80:KEY OFF:COLOR 7,0
11 RANDOMIZE(VAL(RIGHT$(TIME$,2))*60+VAL(MID$(TIME$,4,2)))
12 GOTO 1000
20 DEFSNG A-Z:HG=640:VG=325:HO=79:VE=23:GOTO 1010
100 SCREEN 0:CLS:RETURN
110 IF HO>79 OR VE>23 THEN RETURN
111 LOCATE VE+1,HO+1,1:RETURN
120 HO=POS(0)-1:VE=CSRLIN-1:RETURN
150 COLOR 0,7
151 PRINT "   "+SR$+"   "
153 COLOR 7,0
155 RETURN
200 IN$=INKEY$
201 IF IN$="" THEN IN=0:GOTO 209
202 IN=ASC(IN$):IF IN>96 AND IN<123 THEN IN=IN-32
203 IF IN<>0 THEN 209
204 IN=ASC(RIGHT$(IN$,1))
205 IF (IN=75) OR (IN=77) THEN IN=(IN-19)/2
206 IF (IN=80) OR (IN=72) THEN IN=40-IN/8
207 IF (IN=83) THEN IN=127
209 RETURN
210 GOSUB 200
211 IF IN$=""THEN 210
212 RETURN
250 BEEP:RETURN
260 RV=RND(1):RETURN
270 FR=FRE(0)+FRE(""):RETURN
300 SR$=MID$(STR$(SR),2+(SR<0)):RETURN
310 O#=ABS(SR)+.5*10^-CN:O!=INT(O#):O#=1+O#-O!:O$=""
311 IF CN THEN O$="."+STRING$(CN,48):IF O#>1 THEN MID$(O$,1)=MID$(STR$(O#),3)
312 O$=MID$(STR$(O!),2)+O$:IF VAL(O$)>0 AND SR<0 THEN O$="-"+O$
313 IF LEN(O$)<=CT THEN SR$=SPACE$(CT-LEN(O$))+O$:RETURN
314 SR$=STRING$(CT,42):RETURN
330 FOR O1=1 TO LEN(SR$)
332 O2$=MID$(SR$,O1,1)
334 IF (O2$<"a") OR (O2$>"z") THEN 340
336 MID$(SR$,O1,1)=CHR$(ASC(O2$)-32)
340 NEXT O1
341 RETURN
350 LPRINT SR$;:RETURN
360 LPRINT:RETURN
400 IF SP>119 THEN SP=119
405 IF SP<36 THEN SP=36
406 IF SD>20 THEN SD=20
410 PLAY "L"+STR$(INT(20/SD))+"N"+STR$(SP-35)
420 RETURN
500 IF INT(NF/2)*2<>NF THEN 510
504 OPEN "I",NF,NF$
508 RETURN
510 OPEN "O",NF,NF$
516 RETURN
540 IN$=""
542 IN=0
544 IF EOF(NF) THEN IN=1:RETURN
546 LINE INPUT #NF,IN$
548 IF EOF(NF) THEN IN=1
550 RETURN
560 PRINT #NF,SR$
562 IN=0
564 RETURN
580 IN=0
582 CLOSE #NF
584 RETURN
600 SCREEN 105:CLS:RETURN
620 GOSUB 655:IF CN=0 THEN PSET(OH,OV) ELSE PRESET(OH,OV)
621 RETURN
630 GOSUB 655:IF CN=0 THEN LINE-(OH,OV) ELSE LINE-(OH,OV),0
631 RETURN
650 LOCATE 1+INT(VE*25),1+INT(HO*80):PRINT SR$:RETURN
655 OH=HO*HG:IF OH<0 OR OH>HG THEN OH=(-HG+1)*(OH>90)
656 OV=VE*VG:IF OV<0 OR OV>VG THEN OV=(-VG+1)*(OV>90)
657 RETURN
950 GOSUB 100:STOP
961 '
962 '   Einde voorloper voor IBM kloon met superresolutiescherm.
963 '   Veel andere types willen VG=200 in regel 20 en
964 '   SCREEN 2 in regel 600.
965 '
```

Eeuwkalender loopt nog geen dag!

Franciscus Schreiner uit Leiden is ten einde raad. Ook uitgeverij Kluwer en de importeur van Schneider konden zijn eeuwkalender niet aan de praat krijgen. Misschien lukt het ons ook niet hem te helpen, maar hij zou eens kunnen proberen om het volgende, gewijzigde programma in te typen. Werkt het dan nog niet, dan moet er meer informatie komen over wat er op het scherm gebeurt nadat de complete datum ingevoerd is.

```
100 rem eeuwkalender
110 dim d$(6)
120 for i= Ø to 6
130 read d$(i)
140 next i
150 input "dag:";d
160 input "maand:";m
170 input "jaar:";j
180 print
190 if m>3 then goto 220
200 j=j-1
210 m=m+12
220 e=int((13/5)*(m+1))
230 f=int(5*j/4)
240 g=int(j/1ØØ)
250 h=int(j/4ØØ)
260 t=d+e+f-g+h
270 da=int(((t/7)-int(t/7))*7+Ø.1)
280 print"die dag viel op ";d$(da);"dag"
290 data"zater","zon","maan","dins"
300 data"woens","donder","vrij"
310 end
```

Samenvoegen van programmadelen

J. Wolters Aben wil graag in contact komen met andere BASICODE gebruikers om programma's uit te wisselen. Het adres staat hieronder. Maar er is nog een ander probleem dat waarschijnlijk allang opgelost was als dat contact met andere gebruikers er al was geweest. Op zondag 10 mei werd er door Hobbyscoop een programma over woordontleding in twee delen uitgezonden. Vooral die programma's over taal en rekenen vallen bij de kinderen Wolters erg in de smaak. De vraag is echter hoe die twee delen aan elkaar gekoppeld moeten worden. In vakjargon heet de opdracht voor het aan elkaar koppelen MERGE of APPEND. Ook bij BASICODE is het mogelijk delen aan elkaar te koppelen. Dat gaat als volgt:

- Laad eerst het eerste deel op de normale manier.
- Laad nu het tweede deel er achteraan met de opdracht voor BIJ-laden.

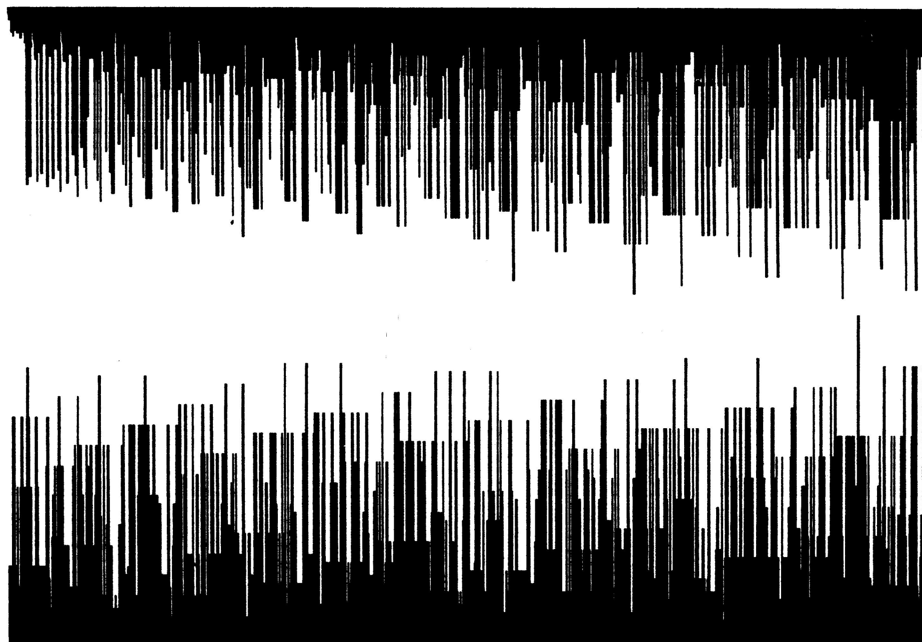
Alles wat het eerst ingeladen is, blijft bewaard en het tweede stuk komt er gewoon achter te staan.
Elk vertaalprogramma heeft zo'n opdracht voor bijladen, het laden van een deel zonder de regels boven 1000 te wissen; zie daarvoor het BASICODE-boek of lees de handleiding in het vertaalprogramma zelf eens goed. Op deze manier kunnen ook meer dan twee stukken aan elkaar gekoppeld worden. Dat kan wel eens handig zijn om eerder gemaakte routines opnieuw te gebruiken in een programma dat u aan het maken bent.
Nu vraagt u zich misschien af waarom men programma's in stukken uitzendt. Welnu dat zit hem in de maximale lengte van programma's die sommige computers kunnen verwerken. Een programma in BASICODE is namelijk een zogenaamde ASCII-file. Alle tekens zoals die in de listing van het programma te zien zijn, worden uitgezonden. In de computer gaat dat anders. Daar is voor elk BASIC woord een enkel teken (ook wel token genoemd) in gebruik (de vijf letters van het woord PRINT bijvoorbeeld, worden vervangen door een teken). De daarvoor gebruikte tekens komen in de ASCII-tabel (met gewone lettertekens, cijfers en leestekens) niet voor en verschillen bovendien per computertype. Het voordeel is echter dat er veel minder ruimte wordt ingenomen in het geheugen. Het nadeel is dat voor het zichtbaar maken van de listing op het scherm een omzetting nodig is in de computer. Als het eerste deel van een lang programma is geladen en omgezet in de eigen tekens van de computer, neemt dat deel dus minder ruimte in. Het is goed mogelijk dat het tweede deel er nu bij past in het geheugen terwijl dat bij het laden in een keer niet gelukt zou zijn.
Neemt u nog even contact op met onze briefschrijven(ster) voor de uitwisseling van BASICODE-programma's?
Adres: Stationsweg 28
6075 CC Herkenbosch tel: 04752-2558

Mannetjes of vrouwtjes?

Voor de computer maakt het natuurlijk helemaal niets uit of er nu een mannetje of een vrouwtje meewerkt. Of zou het voor het toetsenbord toch verschil uitmaken? Voor de redactie is het echter prettig te weten of een brief van een man of een vrouw afkomstig is omdat we bij het aanspreken van de persoon graag de juiste vorm gebruiken. Laten we afspreken dat u ons daarbij voortaan een beetje helpt. (De redactie bestaat op dit moment uit mannetjes.)

Trapgevels in Syracuse?

De geheimen van Syracuse blijven de lezers boeien. M. Keuper uit Enschede heeft weer een nieuw licht geworpen op de beruchte Syracuse-rijen. De heer Keuper heeft het programma van W. Heijbroek uit Oudkarspel als uitgangspunt genomen en is daarmee op de ATARI 1040 van zijn moeder aan de slag gegaan. Daarna heeft hij het veranderd zodat er geen GOTO's meer in voor kwamen. Hij heeft daarbij geen gebruik gemaakt van de truc die ik in het vorige artikel voorstelde maar is met



❶

GFA-basic aan de slag gegaan. Fraai, maar onbruikbaar voor anderen die deze basic niet bezitten. Het programma drukt van de oneven termen de aantallen af op grafische wijze. Op het plaatje (figuur 1) ziet u op de bovenste rij de oneven getallen van 1 tot en met 1280 en op de onderste rij die van 1280 tot en met 2560.
Als u het papier op een afstandje bekijkt, of er met half gesloten ogen naar kijkt, ziet u een soort trapsgewijze oploop van aantallen termen. Een prima idee van de heer Keuper, dat grafisch weergeven. Vaak zie je op deze manier veel sneller waar de oplossing gezocht moet worden.

❷

```
For F=1 To 2560 Step 2
  G=F
  L=0
  For K=0 To 1
    L=L+1
    H=G/2
    If H=Int(H)
      G=G/2
    Else
      G=G*3+1
    Endif
    If G>1
      K=0
    Endif
  Next K
  If F<1280
    Line F/2,0,F/2,L
  Else
    Line F/2-640,400,F/2-640,400-L
  Endif
Next F
```

Het keurige programmain GFA-basic staat in figuur 2. Het is heel gemakkelijk om te zetten naar BASICODE-3 waarbij in ieder geval de lijntjes op het scherm zichtbaar gemaakt kunnen worden. Wie maakt voor de volgende keer een BASICODE-3 programma (kijk ook nog eens even naar de vorige artikelen over dit onderwerp) dat net zo netjes is (zonder GOTO's) en op het scherm de trapsgewijze oploop zichtbaar maakt? Wellicht is op de een of andere manier wel een deel van de lijntjes weg te filteren zodat het beeld duidelijker wordt.

Gratis vertaalprogramma voor C-64

Als auteur van het Commodore-64 vertaalprogramma ben ik Rien van Dongen natuurlijk erg dankbaar voor de suggesties in het artikel over de cijferagenda. Daardoor heb ik nu de bestandsroutines zo kunnen maken als ze in BASICODE-3 zijn bedoeld. Inmiddels had ik ook al wat andere foutjes verbeterd, zodat er thans een aardig bijgewerkte nieuwe versie beschikbaar is, die bij de eerstvolgende druk van het BASICODE-3 boek op de cassette zal worden opgenomen. Het BASICODE-3 boek is overigens gewoon in de boekhandel te koop. Voor de lezers van dit blad heb ik een verrassing, daarover direct meer. Laat ik beginnen met uit te leggen waarom er in het C-64 vertaalprogramma nog fouten zitten.
Bij vertaalprogramma's hebben we te maken met een erg ingewikkelde toestand en ze zijn daarom grotendeels in machinetaal geschreven. Voor de C-64 beslaat het vertaalprogramma liefst 53 pagina's. Zoiets kan per definitie niet foutloos. Ook in het BASICODE-2 vertaalprogramma zitten nog steeds enkele fouten. Bovendien heb ik zelf geen C-64 (nooit gehad ook) en ik moest het vertaalprogramma dus ontwikkelen op een geleende machine.
Nu de verrassing. Lezers van dit blad die een lege datacassette hebben, zoals die geleverd is bij het BASICODE-3 boek, kunnen als eerste beschikken over de nieuwste versie van het vertaalprogramma voor de C-64. Daarvoor moet u als volgt handelen:

- Neem het paginanummer uit A&K-Informatica.
- Doe de lege datacassette in een STEVIGE maar OPEN retour-envelop.
- Schrijf op die open retourenvelop uw EIGEN naam en adres en frankeer die voldoende (even wegen!).
- Stop nu een stevige open envelop met de cassette erin samen met het paginanummer in een gewone envelop.
- Frankeer deze gewone envelop ook weer voldoende (alles samen wegen!) en zet er onderstaand adres op.
- Controleer nogmaals of de cassette goed staat, netjes aan het begin, of de binnenste envelop open is en of het paginanummer is ingesloten. Als alles in orde is, kan de buitenste envelop dicht en kan de zending gepost worden.

Ik zet de nieuwe versie precies op de plek waar de cassette staat. Daarna stop ik de cassette terug in de door u geadresseerde en gefrankeerde retour-envelop en stuur die naar u.
Hebt u een kennis die ook de nieuwste versie wil? Help dan elkaar, dat scheelt u aan porto en mij aan werk. Als er overigens een C-64 gebruikersclub bestaat die wil meewerken aan de verspreiding van het nieuwste BASICODE-3 vertaalprogramma, dan kan deze eveneens met mij contact opnemen. Jacques Heubrich
Stichting Basicode,
Postbus 1410, 5602 BK Eindhoven.

Magnetische koeling

Water kunnen we kouder maken door er ijsblokjes in te gooien. Het ijs heeft als vaste stof een hoge mate van ordening, de deeltjes in het vloeibare water zijn helemaal vrij in hun beweging. Het toevoegen van de orde van het ijs beperkt die vrijheid iets: het water wordt kouder.
Een vaste stof is één vorm van ordening in de natuur, een magneetveld is een andere vorm. Ook die orde kan gebruikt worden om dingen te koelen. Dat gebeurt in deze magnetische koelinstallatie, die beproefd wordt door de Hughes Aircraft Company in El Setgundo, Californië. De toepassingen liggen in de ruimtevaart; soms zijn aan boord van satellieten namelijk lage temperaturen nodig.
IRAS was een voorbeeld van een kunstmaan waarvoor koeling onontbeerlijk was. Bij een hogere temperatuur was de satelliet niet in staat geweest de infraroodstraling te detecteren van talloze bronnen in het heelal. Het koelen gebeurde hier met meegenomen vloeibaar helium; toen dat op was, moest de satelliet zijn waarnemingen staken. Magnetische koeling kan dit soort kunstmanen in de toekomst een langer leven verzekeren. Die magnetische koeling wordt dan toegepast samen met het klassieke koelsysteem.
Voor astronomische waarnemingen wordt het koelsysteem echter niet in de eerste plaats ontwikkeld. Ook het Star Wars-project heeft belang bij koele satellieten. Daarbij wordt de vijand via infrarood opgespoord en wordt elektronica toegepast, die een lage temperatuur moet hebben om snel kleine signalen te kunnen verwerken.
Voor de verdere toekomst denkt men ook aan toepassingen op Aarde. Het gaat dan om gebruik in elektriciteitsnetten, waar koeling nodig kan worden, wanneer men met supergeleiding gaat werken. (W.v.T.)

Rien van Dongen
Siso code 365.2

Cijferagenda

beslissingsondersteuning op school

Schoolcijfers vormen het schrikbeeld van vele leerlingen. En hoewel ook docenten er een hekel aan hebben, zijn cijfers haast onmisbaar bij de beoordeling van leerprestaties. Rapporten afschaffen? Onzin, ook de leerlingen zelf willen weten hoe ze er voor staan en vergelijken hun prestaties graag met medeleerlingen. Cijfers afschaffen? Ook onzin! Elke uitdrukking over de prestatie laat zich even gemakkelijk uitdrukken in cijfers als in woorden. Prestatie afschaffen? Natuurlijk niet, juist competitie kan de samenleving op een hoger plan brengen. Maar geen afvalrace zoals in de sport. Een leerling met een twee moet na het behalen van een acht weer gewoon meespelen.

Rechtvaardig rekenen

De brief van Eddy Kroes, leraar aan de MTS in Terneuzen, was de aanleiding voor dit artikel. Hij vroeg me eens na te denken over een computerprogramma voor de opslag en verwerking van cijfers. Omdat ik al vaker, ook bij het geven van cursussen aan docenten, met deze vraag werd geconfronteerd, besloot ik dit probleem eens nader te bekijken. Eddy wil zo rechtvaardig mogelijk rekenen. Korte overhoringen mogen niet zo zwaar tellen als werkstukken waarin avonden werk zit. Te moeilijke repetities telt hij niet mee. Zieke leerlingen mogen hun werk later inhalen. Het uitrekenen van eindcijfers per leerling en gemiddelden per klas is nogal veel werk als dat in de docentenagenda moet gebeuren. Even uitproberen hoe de cijfers veranderen als een repetitie niet wordt meegeteld, is er niet bij. En dan dat telkens weer overschrijven van de cijfers op rapportlijsten. Genoeg redenen om aan een computerprogramma te denken.

Ruitjespapier

De vorm van een cijferagenda voldoet goed. Voor de cijferadministratie zijn die bladzijden met hokjes heel geschikt. In de eerste kolom staan onder elkaar de namen van de leerlingen en in de volgende kolommen de behaalde cijfers per repetitie of werkstuk. De gemiddelden per rij leveren de eindcijfers per leerling op en in een kolom krijgen we een idee van de prestaties, per repetitie van de klas als geheel. Boven elke kolom met cijfers zetten we echter het gewicht, het getal waarmee we alle cijfers van die repetitie willen vermenigvuldigen. De berekening is nu ineens veel lastiger. En om te kunnen zien hoe de cijfers beïnvloed worden door het al dan niet meetellen van een repetitie, is naast het ruitjespapier, een zakrekenmachine en een heleboel geduld, ook nog een groot vlakgom nodig...

Agenda met een venster

Hoe gebrekkig het ruitjesvel dan ook is bij het berekenen van eindcijfers, het geeft wel overzicht over alle leerlingen en repetities. En voor het wijzigen van een cijfer kunnen we direct het betreffende hokje wissen en er wat anders in schrijven.
Bij een computer ligt dat anders. Het scherm kan alle gegevens van een klas niet in een keer weergeven. Vaak zijn er per regel maar 40 of 80 tekens en niet meer dan 25 regels. Als het ruitjesvel in de computer zit zien we daarvan maar een stukje op het scherm. Het scherm is als het ware een venster waardoor we maar een deel van het binnenste zien. Door het venster of het ruitjesvel te verplaatsen, worden de andere delen zichtbaar.
Met schuiven kunnen we elk cijfer wel op het scherm krijgen, maar uitgummen is minder eenvoudig. Op de een of andere manier moeten we met het toetsenbord (of een computermuis) aangeven waar op het scherm we iets willen wijzigen. We verplaatsen daarbij het (knipperende) aanwijsblokje (-streepje of -pijltje): de cursor. De wijziging zelf moet ook via het toetsenbord gebeuren.
Communicatie van de gebruiker met het ruitjesvel dat in de computer zit, gaat met behulp van apparaten. Het toetsenbord en beeldscherm vormen samen de tussenschakeling: de 'interface'. Zo'n interface moet twee talen spreken.
Een andere interface is die waarmee een printer op de computer aangesloten is. Het is een vernuftige, elektronische schakeling. De signalen van de computer worden geschikt gemaakt voor de printer en omgekeerd stuurt de printer via de interface signalen naar de computer, bijvoorbeeld om het tempo te regelen. Een tussenschakeling voor menselijke gebruikers (user-interface) werkt met symbolen, bijvoorbeeld de letters op het scherm en op de toetsen. Op een beeldscherm kort en eenduidig aangeven wat een gebruiker moet doen, is niet zo eenvoudig. Daarom gebruikt men tegenwoordig vaak de door Xerox Computer uitgevonden Iconen, kleine plaatjes waarop een functie getekend is. Door zo'n plaatje aan te wijzen met de cursor, wordt de betreffende functie ingeschakeld. Overbekend is het plaatje van een prullenbakje dat aangeeft dat gegevens worden gewist.

Probleem analyse

Voor het computerprogramma ligt er nu al een aantal eisen op tafel. Alle namen en cijfers moeten er niet alleen in gezet kunnen worden, ook wijzigen moet mogelijk zijn. Het opslaan en weer terughalen van gegevens met cassette of diskette, moet ook kunnen. En het afdrukken op een printer is een logische eis. Over het schuiven van het venster hebben we het al gehad maar daar zit nog een extra probleem aan vast. Als het venster over het ruitjesvel schuift, zijn telkens bepaalde rijen of kolommen niet zichtbaar. Met het venster helemaal links zien we wel de namen maar niet de kolom met eindcijfers. En staat het venster bovenaan, dan zijn de gewichten wel zichtbaar maar de gemiddelden niet. We willen minstens de bij elkaar behorende delen zowel van links en rechts als boven en onder tegelijkertijd op het scherm zien. Dan is er op het scherm nog informatie nodig ten behoeve van de user-

interface zodat aan de presentatie op het scherm nogal wat eisen worden gesteld. Het programma moet ook rekenen want daarvoor maken we nu juist een computerprogramma. Cijfers worden met een 'gewogen' gemiddelde bepaald. Een 10 met gewicht 2 en een 5 met gewicht 3 leveren een eindcijfer op van 7 (2 x 10 + 3 x 5 gedeeld door 2 + 3): Bij ziekte of andere geoorloofde afwezigheid is er geen cijfer en telt het ook niet mee. Anders wordt een niet gemaakte repetitie wel meegeteld. En een nul haalt het gemiddelde flink omlaag! Het gemiddelde van de hele klas voor een repetitie is een gewoon gemiddelde, maar als een leerling van school is gegaan, mag de betreffende rij met cijfers niet meer meetellen. En ten slotte willen we gemakkelijk even kunnen kijken wat het resultaat is, als we voor de hele klas een "slecht gemaakte" repetitie niet meetellen. Uit deze analyse is direct een schetsontwerp af te leiden.

Rekenblad als prototype

Wellicht denkt u intussen dat het dwaas is zo'n programma te gaan maken terwijl er prima spreadsheets (rekenbladprogramma's) te koop zijn. Misschien is dat ook wel zo. Deze computertoepassing zou ook in een spreadsheet gestopt kunnen worden. Maar een spreadsheet moet wel voorbereid worden. Om aan de genoemde eisen te kunnen voldoen, moet u al aardig met een spreadsheet overweg kunnen. Met name de ingewikkelde presentatie van het ruitjesvel op het scherm is ook in een spreadsheet niet echt gemakkelijk te programmeren. Een spreadsheetprogramma werkt bovendien slechts op één type computer. Maar nu gaan we een BASICODE-programma maken, nog klein van omvang ook, zodat het geschikt is voor nagenoeg elke computer. De vormgeving van spreadsheets heeft wel als voorbeeld, 'prototype', gediend voor het programma. Zo'n prototype is een geweldig hulpmiddel bij het ontwerpen van een computerprogramma. Het is veel gemakkelijker van een werkend prototype te zeggen wat er goed en slecht aan is, dan van een papieren schetsontwerp, zeker voor een opdrachtgever die leek is op computergebied.

Definitief ontwerp

Na het schetsontwerp en de opmerkingen bij het prototype is het mogelijk een definitief ontwerp te maken. Net als een spreadsheet is het helemaal op de gebruiker gericht. De user-interface vormt daarom het hart van het programma, het hoofdprogramma. Omdat we bij het programmeren stapsgewijs willen verfijnen, top-down programmeren, verwijst het hoofdprogramma naar subroutines waarin het eigenlijke werk wordt gedaan.

In figuur 1 staat een programma structuur diagram (PSD) voor het programma.

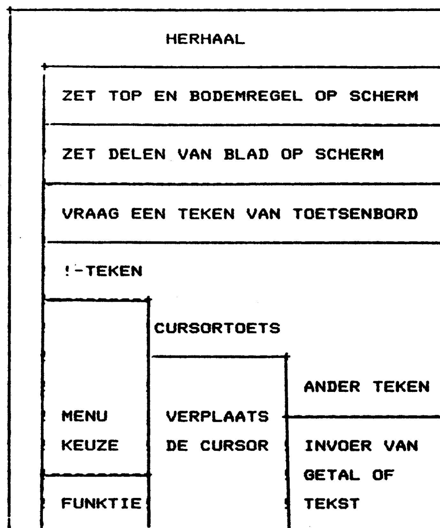


fig. 1: Hoofdprogramma van Cijferagenda.

Bij het menu moeten de volgende mogelijkheden op het scherm verschijnen:
-VUL, het invullen van het blad
-STOP, uitrekenen, opbergen en einde
-LAAD, binnenhalen van een blad
-DRUK, afdrukken op de printer
-REKEN, alleen uitrekenen

Na de keuze wordt de betreffende subroutine ingeschakeld. Bij 'VUL' hoeft die subroutine niets te doen en kan het hoofdprogramma weer bovenaan in het diagram beginnen; (HERHAAL betekent dat het hoofdprogramma steeds opnieuw begint als we onderaan in het diagram zijn aangekomen).

De subroutine die de cursor moet verplaatsen, heeft nog een functie. De cursor wordt natuurlijk binnen het venster gehouden, maar zodra de gebruiker probeert daar buiten te sturen, veranderen de coördinaten van de linkerbovenhoek van het af te drukken cijferblad. Bij de volgende doorgang van het hoofdprogramma wordt het blad daadwerkelijk opgeschoven.

Ook de invoerroutine heeft iets bijzonders. De in te voeren vakjes zijn niet allemaal even groot: een naam is langer dan een getal. Behalve dat die instelling automatisch moet gebeuren, moeten getallen ook nog op een andere manier worden afgedrukt om ze netjes onder elkaar te krijgen. Het spreekt voor zich dat de invoerroutine automatisch naar het begin van de vakjes moet springen. Als er al wat in een vakje stond, kan dat gewijzigd worden. Daarbij zijn de cursortoetsen en de wistoets van belang. We komen echter pas in de invoerroutine nadat er een letter (spatie) of cijfer is ingetypt.

Afsluiten van de invoer gebeurt gewoon met de invoertoets (RETURN). Drukken we op de invoertoets zonder iets te wijzigen, dan blijft het aanwezige gegeven behouden. De problemen van het ontwerp zitten niet in de structuur, die is eigenlijk heel eenvoudig. De details maken het ontwerp lastig.

Programmering

Top-down programmeren is bij zo'n eenvoudige structuur heel gemakkelijk. Toch hebben kleine verfijningen veel tijd gekost. Vaak was dat een gevolg van domme fouten. Vooral de relatie tussen een positie op het scherm, het berekenen van de coördinaten en het daaruit afleiden van de af te drukken strings is een bron van kleine foutjes waardoor er op het scherm de vreemdste dingen kunnen gebeuren. Het programma is zo gemaakt dat er andere instellingen mogelijk zijn. Zo kan de grootte van het rekenblad aangepast worden, bijvoorbeeld aan meer leerlingen of minder cijfers. Ook kan de breedte van de cijfers gewijzigd worden. Eddy Kroes wilde 3 vakjes per cijfer om geen komma in te hoeven typen (een tien wordt 100 en een twee 20). Maar ook een breedte van twee of vier is in te stellen. Afhankelijk van de breedte van de regels op het scherm van uw computer krijgt u automatisch een kleiner of groter deel van het rekenblad te zien. En natuurlijk kunt u overschakelen van diskette naar cassette voor de opslag van de gegevens. Dat gebeurt allemaal vooraan in het programma, bij de initialisatie (1000-2000). Ook dit programma zelf bevat weer geen enkele GOTO-opdracht (behalve die ene die verplicht is bij BASICODE op regel 1000). Daardoor is het heel eenvoudig om te zetten naar welke andere computertaal dan ook.

Het programma is geschreven in BASI-

cijferagenda

klas:vwo 3b vak:	a	b	c	d	
meetellen(0=nee):	0	1	1	1	
gewicht:	1	2	1	1	
1 ab anders	40	67	45	---:	60
2 bert bussem	40	78	67	---:	74
3 cris carelsen	45	0	89	---:	89
4 dik donders	60	67	78	---:	71
5 erik engels	30	87	99	---:	91
6 francine fles	70	0	92	---:	92
7 gonnie garten	40	65	91	---:	74
8 helma haag	34	98	90	---:	95
9 ingrid iepen	20	78	56	---:	71
10 jolanda jansen	56	100	78	---:	93
11 klaas klomp	78	67	45	---:	60
12 leo linders	56	93	84	---:	90
13 marie molen	55	83	94	---:	87
14 nellie noor	34	82	73	---:	79
15 olga olivier	56	71	57	---:	66
16 peter panda	78	78	87	---:	81
17 quirinus quant	78	67	67	---:	67
18 rien rozenboom	67	91	56	---:	79
klas gemiddelde:	54	76	75	---	76

stuur met cursortoetsen (!=menu)

fig. 3: Op het scherm staat maar een gedeelte van het blad met cijfers.

cijferagenda

klas:vwo 3b vak:	c	d	e	f	
meetellen(0=nee):	1	1	1	1	
gewicht:	1	1	1	1	
9 ingrid iepen	56	---	---	---:	71
10 jolanda jansen	78	---	---	---:	93
11 klaas klomp	45	---	---	---:	60
12 leo linders	84	---	---	---:	90
13 marie molen	94	---	---	---:	87
14 nellie noor	73	---	---	---:	79
15 olga olivier	57	---	---	---:	66
16 peter panda	87	---	---	---:	81
17 quirinus quant	67	---	---	---:	67
18 rien rozenboom	56	---	---	---:	79
19 stef staal	78	---	---	---:	82
20 tinus tent	45	---	---	---:	52
21 udo den uil	78	---	---	---:	71
22 era van veen	67	---	---	---:	---
23 wilma welstand	56	---	---	---:	70
24 xavira xantus	78	---	---	---:	71
25 yvonne ypestyn	98	---	---	---:	70
26 zacharias zon	78	---	---	---:	56
klas gemiddelde:	75	---	---	---	76

stuur met cursortoetsen (!=menu)

fig. 4: Leerling 22 doet niet meer mee, de naam begint met een spatie.

fig. 2: De gewijzigde routines voor de C-64.

```
500 in=0:x%=nf:ifx%<2thensys49185:return

501 ifx%<4thenopenx%,1,x%-2,nf$:return
502 open15,8,15:print#15,"i":if(x%and1)=
0thenopenx%,8,x%,"b"+nf$+",s,r":goto 504

503 openx%,8,x%,"@0:b"+nf$+",s,w"
504 input#15,in,o$,o$,o$:in=sgn(abs(in))
:return
540 ifnf<2thensys49188:return
541 input#nf,in$:in=sgn(abs(st)):return
560 ifnf<2thensys49191:return
561 print#nf,chr$(34);sr$;chr$(13);chr$(
13);:in=sgn(abs(st)):return
580 x%=nf:ifx%<2thensys49194:return
581 closex%:in=0:print#15,"i":close15:re
turn
```

klas:vwo 3b	vak:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	
	meetellen(Ø=nee):	Ø	1	1	1	1	1	1	1	1	1	1	1	1	1	:
	gewicht:	1	2	1	1	1	1	1	1	1	1	1	1	1	1	:
1	ab anders	4Ø	67	45	---	---	---	---	---	---	---	---	---	---	---:	6Ø:
2	bert bussem	4Ø	78	67	---	---	---	---	---	---	---	---	---	---	---:	74:
3	cris carelsen	45	Ø	89	---	---	---	---	---	---	---	---	---	---	---:	89:
4	dik donders	6Ø	67	78	---	---	---	---	---	---	---	---	---	---	---:	71:
5	erik engels	3Ø	87	99	---	---	---	---	---	---	---	---	---	---	---:	91:
6	francine fles	7Ø	Ø	92	---	---	---	---	---	---	---	---	---	---	---:	92:
7	gonnie garten	4Ø	65	91	---	---	---	---	---	---	---	---	---	---	---:	74:
8	helma haag	34	98	9Ø	---	---	---	---	---	---	---	---	---	---	---:	95:
9	ingrid iepen	2Ø	78	56	---	---	---	---	---	---	---	---	---	---	---:	71:
1Ø	jolanda jansen	56	1ØØ	78	---	---	---	---	---	---	---	---	---	---	---:	93:
11	klaas klomp	78	67	45	---	---	---	---	---	---	---	---	---	---	---:	6Ø:
12	leo linders	56	93	84	---	---	---	---	---	---	---	---	---	---	---:	9Ø:
13	marie molen	55	83	94	---	---	---	---	---	---	---	---	---	---	---:	87:
14	nellie noor	34	82	73	---	---	---	---	---	---	---	---	---	---	---:	79:
15	olga olivier	56	71	57	---	---	---	---	---	---	---	---	---	---	---:	66:
16	peter panda	78	78	87	---	---	---	---	---	---	---	---	---	---	---:	81:
17	quirinus quant	78	67	67	---	---	---	---	---	---	---	---	---	---	---:	67:
18	rien rozenboom	67	91	56	---	---	---	---	---	---	---	---	---	---	---:	79:
19	stef staal	76	84	78	---	---	---	---	---	---	---	---	---	---	---:	82:
2Ø	tinus tent	56	55	45	---	---	---	---	---	---	---	---	---	---	---:	52:
21	udo den uil	54	67	78	---	---	---	---	---	---	---	---	---	---	---:	71:
22	era van veen	89	88	67	---	---	---	---	---	---	---	---	---	---	---:	---
23	wilma welstand	23	77	56	---	---	---	---	---	---	---	---	---	---	---:	7Ø:
24	xavira xantus	45	67	78	---	---	---	---	---	---	---	---	---	---	---:	71:
25	yvonne ypestyn	45	56	98	---	---	---	---	---	---	---	---	---	---	---:	7Ø:
26	zacharias zon	23	45	78	---	---	---	---	---	---	---	---	---	---	---:	56:
27	angelien alias	67	78	87	---	---	---	---	---	---	---	---	---	---	---:	81:
28	belinda baars	76	9Ø	78	---	---	---	---	---	---	---	---	---	---	---:	86:
29	chris crol	65	89	67	---	---	---	---	---	---	---	---	---	---	---:	82:
3Ø	dolly droog	54	78	56	---	---	---	---	---	---	---	---	---	---	---:	71:
31	erika enkel	74	48	78	---	---	---	---	---	---	---	---	---	---	---:	58:
32	flip flop	83	9Ø	85	---	---	---	---	---	---	---	---	---	---	---:	88:
33	greet graspol	54	93	96	---	---	---	---	---	---	---	---	---	---	---:	94:
34	hans hofnar	34	87	84	---	---	---	---	---	---	---	---	---	---	---:	86:
35	irma ieriks	56	78	75	---	---	---	---	---	---	---	---	---	---	---:	77:
36	jan janssens	65	58	78	---	---	---	---	---	---	---	---	---	---	---:	65:
37	kas koster	47	74	67	---	---	---	---	---	---	---	---	---	---	---:	72:
klas	gemiddelde:	54	76	75	---	---	---	---	---	---	---	---	---	---	---	76:

fig. 5: Een uitdraai op de printer geeft een overzicht. De vakken zijn hier aangeduid met letters maar dat mogen ook woordjes zijn.

CODE-3 omdat opslag van gegevens nodig was. Daarvoor dienen de routines van uw vertaalprogramma op de regels 500, 540, 560 en 580. Voorts is de routine op 450 gebruikt om even te wachten. Deze routines kunt u waarschijnlijk zelf toevoegen aan uw BASICODE-2 vertaalprogramma als u nog geen BASICODE-3 programma hebt. Een probleem kan de routine 210 opleveren. In BASICODE-3 geeft deze routine een knipperende cursor en wacht niet alleen op een letter- of cijfertoets maar kan ook de cursorbesturingstoetsen en de wistoets verwerken. Uit het programma blijkt wat die routines allemaal doen. De inputroutine leest tekens van het beeldscherm. Daarbij is de routine 220 in gebruik. Die is niet zo eenvoudig na te maken. Als u nog geen BASICODE-3 bezit, kunt u beter een andere invoerroutine schrijven.

Werken met cijferagenda

Zodra het programma werkt kunnen we de namen van leerlingen invoeren. Na elke naam springt de cursor een regel omlaag voor de volgende naam. De cursor kan niet uit een invoerveldje ontsnappen voordat de invoertoets is ingedrukt. Zoals al eerder opgemerkt, komt u pas in een invoerveldje nadat u een teken hebt ingetypt. Onderaan het scherm kan de cursor niet meer verder en schuiven alle namen naar boven. Staat in een vakje geen naam, of begint de naam met een spatie, dan telt die leerling niet (meer) mee. Het is dus heel eenvoudig een leerling te verwijderen zonder alle gegevens van die leerling te schrappen. Bij het laatste vakje van deze kolom kunnen we niet verder. We gaan terug naar boven en een kolom naar rechts. Daar komt eerst de naam van de repetitie te staan. Daaronder veranderen we de 1 die er al staat in een 0 als we de repetitie niet mee willen tellen. Vervolgens stellen we het gewicht in en gaan we een voor een de cijfers invoeren. In alle cijferinvoerveldjes worden de getallen automatisch netjes onder elkaar gezet. Het cijfer nul (of helemaal geen cijfer) betekent voor het programma dat de repetitie niet meetelt bij die ene leerling. De leerling kan de repetitie dan eventueel overdoen maar tot er een cijfer ingevoerd wordt, haalt dit gegeven de gemiddelden niet omlaag. Maar het is ook mogelijk dat een leerling er met zijn pet naar gegooid heeft, of dat deze heeft gespiekt. In dat geval geven we een heel laag cijfer, bijvoorbeeld een 1 en dan telt het cijfer wel mee. De functie REKEN wordt gebruikt om te kijken hoe de zaken ervoor staan, bijvoorbeeld als een repetitie niet mee telt, of als een gewicht gewijzigd is. Als er geen wijzigingen meer gewenst zijn, kunnen de berekende eindcijfers op de printer afgedrukt worden. Ook bewaren van alle gegevens is mogelijk. Hiervoor kiezen we de functie STOP. De computer rekent nu eerst alle gemiddelden uit en schrijft daarna de gegevens naar diskette of cassette. Daarna verschijnt de vraag of we echt willen stoppen. Alleen als het antwoord "ja" is, stopt het programma. Lijsten die eenmaal zijn weggeschreven, kunnen weer opgehaald worden met LAAD. Daarbij moet precies dezelfde naam opgegeven worden als bij het wegschrijven! Sommige BASICODE-3 routines schrijven een iets andere naam op de diskette. In de lijst van programma's op de schijf (directory) kan de naam dus afwijken van de in te voeren naam. De eindcijfers van een periode krijgen we door alle repetities buiten die periode niet mee te laten tellen, door de betreffende enen in nullen te veranderen. Binnen een kwartier kan iedereen met dit programma werken. Als we gebruik maken van opslag op cassette, kan iedere docent op de thuiscomputer gegevens aanmaken die op school ingelezen kunnen worden, onafhankelijk van het type computer. Op die manier wordt het samenstellen van rapporten kinderspel. Voor het samenvoegen van de cijfers is wel een eenvoudig programma nodig.

Het eindrapport

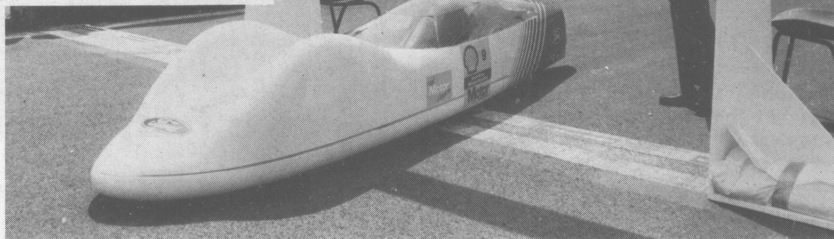
Bij zo'n programma over cijfers hoort een beoordeling. Het programma is getest op een Commodore 64 en op een Acorn BBC. Bij de Commodore zijn de routines voor het opslaan van gegevens op diskette eigenlijk nog niet af (dat staat ook in het BASICODE-3 boek!). Ik heb intussen een aanpassing gemaakt waardoor de foutmeldingen nu wel werken en tevens een andere fout gecorrigeerd. Bij het wegschrijven van strings moeten die (bij de Commodore) tussen aanhalingstekens staan om spaties aan het begin niet te verliezen. Dit programma maakt enkele strings die met een spatie beginnen. Dat ging dus mis maar met extra aanhalingstekens liep het programma goed. Bij het teruglezen worden die aanhalingstekens namelijk automatisch verwijderd. Dit probleem moet echt in de BASICODE-3 routines opgelost worden en niet in het programma. De Acorn BBC bijvoorbeeld leest namelijk de aanhalingstekens wel terug. De regels die u bij de Commodore moet wijzigen, vindt u in figuur 2. Daarin staan ook de andere door mij aangebrachte wijzigingen. Omdat de strings die naar cassette of diskette worden gestuurd bij de Commodore maar 80 karakters lang mogen zijn, is de lengte van de strings in het programma op 78 gezet. Met de twee aanhalingstekens erbij komen we dan op een lengte van 80. Het is eigenlijk heel eenvoudig om de routines van deze computer zo te veranderen dat wel langere strings weggeschreven mogen worden. Maar ik weet niet hoe lang de strings bij alle andere computers mogen zijn.
Bij de BBC kwam er onderaan het scherm een regel te veel te staan, na het gebruik van de cassette. Waarschijnlijk een gevolg van het afdrukken van schermboodschappen door de BASICODE-3 routines. Maar met diskette ging alles goed zolang ik om een bestaande lijst vroeg. Bij het openen van een niet bestaand bestand geeft de computer een foutmelding op het scherm voordat de in het programma ingebouwde foutafhandeling in werking kan treden. Er zijn dus nog wel enige probleempjes bij het werken met bestanden in BASICODE-3. Juist om zulke problemen op te kunnen lossen is het nodig dat u programma's goed test. Waarschijnlijk wordt het programma binnenkort uitgezonden door de TROS, zodat u het niet behoeft over te typen. Stuurt u het testrapport even naar de redactie van dit tijdschrift?

```
995 REM  Hiervoor de subroutines van
996 REM  Basicode-3 toevoegen!
997 REM
1000 A=5000:GOTO 20:REM ..CIJFERAGENDA **
1010 PRINT"Even geduld."
1020 CN=0:CT=3:REM GETALFORMAAT(CT=2-4)
1030 CD=1:REM CASS/DISKETTE (0 OF 1)
1040 HM=HO:REM SCHERM BREEDTE
1050 VM=VE:REM SCHERM HOOGTE
1060 BS=78:REM WERKBLAD BREEDTE
1070 NS=40:REM WERKBLAD HOOGTE
1080 DIM CA$(NS):DIM M$(5):SP$=""
1090 FOR A=0 TO HM:SP$=SP$+" ":NEXT A
1100 BA$="":BB$="":BC$="":CB=CT
1110 R$=LEFT$("---------",CT):S$=""
1120 IF CT>1 THEN S$=LEFT$(SP$,CT-1)
1130 T$=LEFT$(SP$,CB+1)
1200 FOR A=(BS-CT-17)/(CT+1) TO 1 STEP -1
1210 B$=CHR$(64+A)
1220 BA$=" "+B$+S$+BA$
1230 BB$=" "+"1"+S$+BB$
```

```
1240 BC$=" "+R$+BC$
1250 NEXT A
1260 CA$(0)="klas:        vak:"
1270 CA$(1)="meetellen(0=nee):"
1280 CA$(2)="            gewicht:"
1290 CA$(NS)="klas  gemiddelde:"
1300 CT=2:REM TIJDELIJK GETALFORMAAT
1310 FOR A=0 TO NS
1320 SR=A-2:GOSUB 310
1330 SR$=SR$+LEFT$(SP$,15)
1340 IF A=0 THEN A$=BA$
1350 IF(A=1) OR(A=2) THEN A$=BB$
1360 IF A>2 THEN A$=BC$
1370 B$=":"+R$
1380 IF(A<3) OR(A=NS) THEN B$=T$
1390 IF(A>2) AND(A<NS) THEN CA$(A)=SR$
1400 CA$(A)=CA$(A)+A$+B$
1410 NEXT A
1420 CT=CB
1500 M$(0)="VUL  stop laad druk reken"
1510 M$(1)="vul  STOP laad druk reken"
1520 M$(2)="vul  stop LAAD druk reken"
1530 M$(3)="vul  stop laad DRUK reken"
1540 M$(4)="vul  stop laad druk REKEN"
1550 K=(INT((HM-18)/(CT+1))-1)*(CT+1)
1560 E=LEN(CA$(0))
2000 REM ...........WERKEN MET AGENDA **
2010 GOSUB 100:HO=3:VE=4
2020 MK=1:WH=18:WV=0:SH=0:SV=0
2030 FOR AA=1 TO 1
2040 HB=HO:VB=VE
2050 IF MK>0 THEN GOSUB 16000:REM KADER
2060 BB=0:IF(WH<>SH) OR(WV<>SV) THEN BB=1
2070 IF BB=1 THEN GOSUB 13000:REM BLAD
2080 HO=HB:VE=VB:GOSUB 110
2090 MK=0:GOSUB 210
2100 IF IN=33 THEN GOSUB 15000:REM MENU
2110 IF IN>31 THEN GOSUB 19000:REM INPUT
2120 IF MK=1 THEN GOSUB 3000:REM STOP
2130 IF MK=2 THEN GOSUB 4000:REM LAAD
2140 IF MK=3 THEN GOSUB 5000:REM PRINT
2150 IF MK=4 THEN GOSUB 6000:REM REKEN
2160 HO=HB:VE=VB:GOSUB 14000:REM BESTUUR
2170 AA=0
2180 NEXT AA
2190 RETURN:REM -----------------------
3000 REM ......................STOP **
3010 GOSUB 15500:REM NAAM BESTAND
3020 SR$="eerst even rekenen"
3030 VE=VM:GOSUB 17000:REM ACCENTUEER
3040 SD=20:GOSUB 450
3050 GOSUB 6000:REM REKEN
3060 GOSUB 10200:REM STRINGS WEGZETTEN
3070 GOSUB 9000:REM EIND
3080 RETURN:REM -----------------------
4000 REM ......................LAAD **
4010 GOSUB 15500:REM NAAM BESTAND
4020 GOSUB 10000:REM STRINGS LADEN
4030 SH=0:SV=0
4040 RETURN:REM -----------------------
5000 REM .....................PRINT **
5010 GOSUB 12000:REM PRINT
5020 RETURN:REM -----------------------
6000 REM .....................REKEN **
6010 A$=CA$(1):B$=CA$(2)
6020 GOSUB 11000:REM CIJFERS
6030 GOSUB 11500:REM GEMIDDELDEN
6040 GOSUB 13000:REM BLAD
6050 RETURN:REM -----------------------
9000 REM ......................EIND **
9020 SR$="wil je echt stoppen?"
9030 VE=VM:GOSUB 17000:REM ACCENTUEER
9040 GOSUB 210
9050 IF IN=74 THEN GOSUB 950
9090 RETURN:REM -----------------------
10000 REM .............STRINGS LADEN **
10010 SR$="Even geduld!"
10020 VE=VM:GOSUB 17000:REM ACCENTUEER
10030 FC=0:NF=CD*4:GOSUB 500
10040 IF IN<>0 THEN FC=1:GOSUB 18000
10050 IF IN=0 THEN GOSUB 10100
10060 GOSUB 580
10090 RETURN:REM ----------------------
10100 REM .....................LADEN **
10110 FOR BR=0 TO NS
10120 GOSUB 540:CA$(BR)=IN$
10130 IF IN<>0 THEN BR=999
10140 NEXT BR
10160 RETURN:REM ----------------------
10200 REM ..........STRINGS WEGZETTEN **
10210 SR$="Even geduld!"
10220 VE=VM:GOSUB 17000:REM ACCENTUEER
10230 FC=0:NF=CD*4+1:GOSUB 500
10240 IF IN=0 THEN GOSUB 10300
10250 IF IN<>0 THEN FC=2:GOSUB 18000
10260 GOSUB 580
10270 RETURN:REM ----------------------
10300 REM .................WEGZETTEN **
10310 FOR BR=0 TO NS
10320 SR$=CA$(BR)
10330 GOSUB 560:IF IN<>0 THEN BR=999
10340 NEXT BR
10350 IF BR>998 THEN FC=2:GOSUB 18000
10360 RETURN:REM ----------------------
11000 REM ...................CIJFERS **
11010 A$=CA$(1):B$=CA$(2)
11020 VE=VM:B=BS-CT-1
11030 FOR NM=3 TO NS-1
11040 C$=MID$(CA$(NM),4,1)
11050 SR$=R$:IF C$<>" " THEN GOSUB 11200
11060 CA$(NM)=LEFT$(CA$(NM),B)+SR$
11070 SR=NS-NM+1:GOSUB 310
11080 SR$="nog "+SR$+" cijfers"
11090 GOSUB 17000:REM ACCENTUEER
11100 NEXT NM
11110 RETURN:REM ----------------------
11200 REM .................MEETELLEN? **
11210 V=0:W=0:TT=0
11220 FOR C=19 TO B STEP CT+1
11230 W=VAL(MID$(CA$(NM),C,CT))
11240 IF W<>0 THEN GOSUB 11300
11250 NEXT C
11270 IF TT>0 THEN:SR=V/TT:GOSUB 310
11280 RETURN:REM ----------------------
11300 REM ........TELLING MET GEWICHT **
11310 F=VAL(MID$(B$,C,CT))
11320 G=F*SGN(VAL(MID$(A$,C,CT)))
11330 V=V+W*G:TT=TT+G
11340 RETURN:REM ----------------------
11500 REM ................GEMIDDELDEN **
11510 A$=""
11520 FOR B=19 TO BS-1 STEP CT+1
11530 SR=(BS-B)/(CT+1)+1:GOSUB 310
11540 SR$="nog "+SR$+" gemiddelden"
11550 GOSUB 17000:REM ACCENTUEER
11560 V=0:W=0:TT=0
11570 FOR NM=3 TO NS-1
11580 C$=MID$(CA$(NM),4,1)
11590 IF C$<>" " THEN GOSUB 11700
11600 NEXT NM
11610 SR$=R$
11620 IF TT>0 THEN SR=V/TT:GOSUB 310
11630 A$=A$+" "+SR$
11640 NEXT B
11650 CA$(NS)=LEFT$(CA$(NS),17)+A$
11660 RETURN:REM ----------------------
11700 REM ........................TEL **
11710 W=VAL(MID$(CA$(NM),B,CT))
11720 IF W<>0 THEN V=V+W:TT=TT+1
11730 RETURN:REM ----------------------
12000 REM ....STRINGS NAAR DE PRINTER **
12020 SR$="staat de printer gereed?"
12030 VE=VM:GOSUB 17000:REM ACCENTUEER
12040 GOSUB 210
12050 IF IN=74 THEN GOSUB 12100
12060 RETURN:REM ----------------------
12100 REM ............................
12110 SR$="alles wordt nu afgedrukt"
12120 GOSUB 17000:REM ACCENTUEER
12130 FOR BR=0 TO NS
12140 IF BR=NS THEN GOSUB 360
12150 SR$=CA$(BR)+":":GOSUB 350:GOSUB 360
12160 NEXT BR
12170 GOSUB 360
12180 RETURN:REM ----------------------
13000 REM ......................BLAD **
13010 HO=0
13120 FOR VE=1 TO VM-2
13130 NM=WV+VE-1:IF VE<4 THEN NM=VE-1
13140 IF VE=VM-2 THEN VE=VM-1:NM=NS
13150 GOSUB 110
13160 PRINT LEFT$(CA$(NM),17);
13170 PRINT MID$(CA$(NM),WH,K);
13180 PRINT RIGHT$(CA$(NM),CT+1);
13210 NEXT VE
13220 SH=WH:SV=WV
13230 RETURN:REM ----------------------
14000 REM ...................BESTUUR **
14010 IF IN=28 THEN HO=HO-1
14020 IF IN=29 THEN HO=HO+1
14030 IF IN=30 THEN VE=VE+1
14040 IF IN=31 THEN VE=VE-1
14050 IF VE<1 THEN VE=1:WV=WV-1
14060 IF VE>VM-3 THEN VE=VM-3:WV=WV+1
14070 IF(VE=2) AND(HO<18) THEN VE=4:HO=0
14080 IF(VE=3) AND(HO<18) THEN VE=1
14090 MA=3:IF VE=1 THEN MA=5
14100 IF HO=17 THEN HO=MA
14110 IF HO=MA+1 THEN HO=18
14120 FOR T=18 TO HM STEP CT+1
14130 IF HO=T+1 THEN HO=HO+CT
14140 IF HO=T+CT THEN HO=HO-CT
14150 NEXT T
14200 AN=17+K-CT
14210 IF HO<MA THEN HO=MA:WH=WH-CT-1
14220 IF HO>AN THEN HO=AN:WH=WH+CT+1
14230 IF WH<18 THEN WH=18
14231 IF VE<1 THEN VE=1:WV=WV-1
14232 IF VE>VM-3 THEN VE=VM-3:WV=WV+1
14233 IF(VE=2) AND(HO<18) THEN VE=4:HO=0
14234 IF(VE=3) AND(HO<18) THEN VE=1
14240 IF WH>=BS-K THEN WH=BS-K-CT-1
14250 IF WV<0 THEN WV=0
14260 IF WV>NS-VM+3 THEN WV=NS-VM+3
14270 RETURN:REM ----------------------
15000 REM ......................MENU **
15020 FOR RR=1 TO 1
15030 SR$=M$(MK)
15040 VE=0:GOSUB 17000:REM ACCENTUEER
15050 SR$="kies met cursor "
15060 SR$=SR$+"(INVOERtoets)"
15070 VE=VM:GOSUB 17000:REM ACCENTUEER
15080 HO=8+MK*5:VE=0:GOSUB 110:GOSUB 210
15090 IF IN=28 THEN MK=MK-1
15100 IF IN=29 THEN MK=MK+1
15110 IF MK<0 THEN MK=4
15120 IF MK>4 THEN MK=0
15130 IF IN=13 THEN VE=1
15140 RR=VE:IN$="":IN=0
15150 NEXT RR
15170 RETURN:REM ----------------------
15500 REM ..............NAAM BESTAND **
15510 SR$="geef de naam van de klas"
15520 VE=VM:GOSUB 17000:REM ACCENTUEER
15530 VB=1:HB=5:GOSUB 19000:HB=3
15540 NF$=MID$(CA$(0),6,BD)
15550 RETURN:REM ----------------------
16000 REM .....................KADER **
16010 SR$="cijferagenda"
16020 VE=0:GOSUB 17000:REM ACCENTUEER
16030 SR$="stuur met cursortoetsen "
16040 SR$=SR$+"(!=menu)"
16050 VE=VM:GOSUB 17000:REM ACCENTUEER
16060 RETURN:REM ----------------------
17000 REM ................ACCENTUEER **
17020 HO=0:GOSUB 110:PRINT LEFT$(SP$,HM);
17030 HO=INT((HM-6-LEN(SR$))/2)
17040 GOSUB 110:GOSUB 150
17060 RETURN:REM ----------------------
18000 REM .............FOUTMELDINGEN **
18030 GOSUB 250
18040 IF FC=1 THEN SR$="niet gevonden"
18070 IF FC=2 THEN SR$="er is iets mis"
18140 VE=VM:GOSUB 17000:REM ACCENTUEER
18150 SD=40:GOSUB 450
18180 RETURN:REM ----------------------
19000 REM .....................INPUT **
19010 BD=14:IF VB=1 THEN BD=7
19020 IF HB>17 THEN BD=CT
19030 VE=VB:HO=HB
19040 FOR PR=1 TO 1
19050 IF(IN<32) OR(IN>90) THEN IN$=""
19055 IF IN=34 THEN IN$=""
19060 IF IN=127 THEN HO=HO-1
19070 IF HO>BD+HB-1 THEN HO=BD+HB-1
19080 IF IN=28 THEN HO=HO-1
19090 IF IN=29 THEN HO=HO+1
19100 IF HO<HB THEN HO=HB
19110 GOSUB 110:PRINT IN$;:GOSUB 120
19120 IF IN=127 THEN PRINT" ";
19130 GOSUB 110:GOSUB 210
19140 IF IN<>13 THEN PR=0
19150 NEXT PR
19160 SR$=""
19170 FOR PR=1 TO BD
19180 HO=HB+PR-1:GOSUB 220
19190 SR$=SR$+CHR$(IN)
19200 NEXT PR
19210 SR=VAL(SR$)
19220 IF(HB>17) AND(VE>3) THEN GOSUB 310
19230 IN$=SR$
19240 NM=WV+VE-1:IF VE<4 THEN NM=VE-1
19250 PO=HB:IF HB>17 THEN PO=PO+WH-18
19260 A$=LEFT$(CA$(NM),PO)+IN$
19270 B$=RIGHT$(CA$(NM),E-PO-BD)
19280 CA$(NM)=A$+B$
19290 HO=HB:GOSUB 110:PRINT IN$;
19293 VB=VB+1
19295 RETURN:REM ----------------------
30000 REM ****************************
30010 REM *        CIJFERAGENDA         *
30020 REM *        IN  BASICODE-3       *
30030 REM *    DOOR: RIEN VAN DONGEN    *
30040 REM *       (C) 11 MEI 1987       *
30050 REM ****************************
```

Aerodynamisch voertuig





De UFO 2 in actie.

Twee jaar geleden heeft Ford-automobielen een uiterst zuinig en supergestroomlijnd wagentje op het Franse Paul Ricard-circuit getest. Het researchvoertuigje heeft de naam UFO 2 gekregen (UFO = Ultimate Fuel Optimiser) en is daarmee een verbeterde versie van de reeds bestaande UFO 1. Het verschil zit hem voornamelijk in een belangrijke gewichtsvermindering van drie kilo (ongeveer 11%).
Het wagentje reed op 1 liter benzine maar liefst 746 kilometer hetgeen echter nog geen wereldrecord is want dat staat op 0,0496 liter per 100 kilometer oftewel 1 : 2016.
De motor, die een cilinderinhoud heeft van 15 cm^3 en een maximum vermogen van 540 Watt bij 5000 toeren per minuut, werd vanuit horizontale positie 15 graden gekanteld om de olie-afvoer te verbeteren.

De luchtweerstandscoëfficiënt (Cw-waarde) is met 0,087 uiterst laag. Normale auto's laten ongeveer 0,3 meten. Een belangrijke opmerking is dat deze Cw-waarde bij 25 graden zijwind 0 wordt en zelfs -0,05 bij 30 graden zijwind, hetgeen betekent dat het wagentje als het ware naar voren wordt gezogen. Men denke hierbij aan hetzelfde effect als bij een zeilboot gebeurt.

De UFO 2 weegt 26 kg en is vervaardigd van Keflar en Nomex honingraatmateriaal. De voorruit bestaat uit Lexan polycarbonaat en is slechts 1 mm dik.
Met dit eenpersoonsautootje won Ford de felbegeerde trofee voor het laagste benzineverbruik op het Paul Ricard-circuit. (R.G.L.)

Volvo's "kop van Jut"



Stefan Nilsson van het veiligheidslab van Volvo met het ongelukkenhoofd-met-sensoren: een veredeld soort kop van Jut dus.

"Wie zijn neus schendt", zegt het spreekwoord, "schendt zijn aangezicht". Hoewel deze wijsheid werd bedacht lang voordat de automobiel zijn intrede deed in het dagelijks straatbeeld, is de juistheid ervan ook achter het stuur volledig toepasbaar. Van de Zweden - befaamd om hun aandacht voor veiligheid - mocht verwacht worden dat zij zich deze feiten zouden aantrekken en inderdaad: de firma Volvo heeft zich dan ook een ware "kop-van-Jut" aangemeten. Deze Jut-kop onderscheidt zich van zijn soortgenoten op de kermis door het bezit van 52 druk-sensoren, die gemaakt zijn van uitermate dun piezo-elektro polyurethaan film. We hebben dus duidelijk van doen met een aanzienlijk kostbaarder voorwerp dan waarop de kermisklant zijn vandalisme legaal mag botvieren.
De klap waarmee het hoofd van een autorijder bij een botsing tegen een onderdeel van de auto slaat, duurt in het algemeen ééntiende seconde. Maar in die tijd kunnen zeer zware verwondingen aan het gezicht ontstaan. Volvo besloot daarom een "sensorenkop" te laten bouwen zodat precies nagegaan kan worden, hoe en waar gelaatsverwondingen optreden om daarna te kunnen uitzoeken hoe de auto ook op dat punt veiliger kan worden gemaakt.
De reden dat men 52 sensoren nodig heeft is gelegen in het feit dat het bot van de schedel verschillende sterktes heeft. Bij de wenkbrauwen bijvoorbeeld, kan de schedel een viermaal zo krachtige klap opvangen als op de bovenkaak. Vroegere onderzoeken op dit gebied gebeurden met een kunststoffilm die van kleur veranderde onder de invloed van druk: Hoe hoger de druk, hoe sterker de kleurverandering. Deze methode was niet alleen erg moeilijk te calibreren, maar men moest bovendien het proefongeluk eerst laten gebeuren en daarna kon men kijken wat er precies gebeurd was. Met Piezo-elektrofilm kunnen de gebeurtenissen geregistreerd worden terwijl ze gebeuren. Daarvoor moet wel een redelijk capabele computer beschikbaar zijn teneinde de metingen vast te leggen. Zo'n apparaat is er uiteraard en hij zit bij tests ingebouwd in de achterbank. Het toestel kan in ééntiende seconde uit de 52 sensoren niet minder dan 800 verschillende metingen doen.
De kop, gemodelleerd uit alluminium, is te koop voor de ronde som van 65.000 Zweedse Kronen, compleet met sensoren en bedrading. Men concentreert zich, zegt Volvo, in de autobouw op maatregelen die het leven van de inzittenden moeten beschermen. Maar men moet verder gaan en ook proberen verwondingen en beschadigingen te voorkomen. (H.L./G.J.)

Dr. W. van Tend

16-megabitchip

Het Japanse bedrijf Nippon Telegraph and Telephone heeft een geheugenchip weten te maken met een opslagcapaciteit van 16 miljoen bits. Geheugenchips van 1 miljoen bits zijn al een paar jaar te koop, terwijl een chip met 4 miljoen bits net op de markt is verschenen. Voordat de 16-megabitchip in de handel komt, zullen nog enige jaren verstrijken.

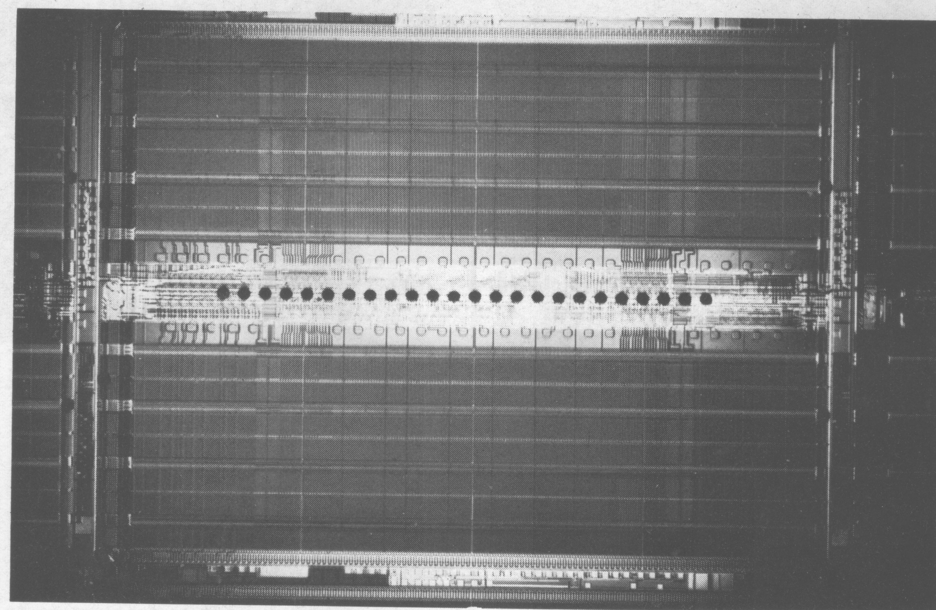
Om één teken (letter of cijfer) op te slaan zijn acht bits nodig. Dat aantal heet één byte. Zestien megabit komt overeen met 2 miljoen bytes, oftewel 2000 kilobytes. Een flinke personal computer heeft een geheugenomvang van 640 kilobyte, die verdeeld is over hele rijen chips. Met de 16-megabitchip wordt driemaal het geheugen van een personal computer ondergebracht in één blokje.

Een Dynamisch RAM

Het gaat hier om een DRAM. RAM staat zoals gebruikelijk voor Random Access Memory, geheugen dat zowel voor lezen als voor schrijven toegankelijk is. De D staat voor Dynamisch. Dat soort geheugen werkt met condensatortjes. Is een condensatortje geladen, dan is het betreffende bit één; is het ongeladen, dan is het nul.
Een dynamisch RAM werkt dus met statische elektriciteit. Het dynamische zit in het feit dat condensatoren lekken. Het gaat hier om heel kleine condensatortjes en die lekken extra erg. De lading op de condensatortjes (de inhoud van het geheugen) moet daarom regelmatig ververst worden. In computers die met DRAM-chips werken, is er dan ook een signaallijn refresh, waarop wordt aangegeven dat het weer eens tijd is de condensatortjes bij te laden.
De kunst van de grote geheugens op één chip is het maken van heel veel, heel kleine condensatortjes. Te klein mogen de condensatortjes ook weer niet zijn: ze bevatten dan zo weinig lading dat ze niet meer kunnen worden uitgelezen.
De klassieke manier is de condensatortjes liggend op de chip aan te brengen. Een condensatortje dat verticaal staat, neemt echter drie maal minder plaats in. IBM werkte in de 1-megabitchip nog met liggende condensatoren, maar ging over op staande voor de 4-megabitchip. Japan maakt zowel 1-, 4- als 16-megabitchips met staande condensatoren. Japan gebruikt de staande-condensatortechniek dus ook, waar het nog niet echt nodig is. Daarom is de indruk in de Verenigde Staten, dat de Japanners deze nieuwe werkwijze beter beheersen.

Japanse overheersing

Niet alleen over een technologische voorsprong van Japan maken de Amerikanen zich zorgen, ook over de verhoudingen op de markt. Van de geheugenchips die in de Verenigde Staten worden toegepast, is 65 procent van Japanse makelij. De wereldmarkt is voor 80 procent in Japanse handen. Inmiddels hebben de Verenigde Staten een groot onderzoeksprogramma voor chiptechnologie gestart, waarbij veel aandacht wordt gegeven aan geheugens. De Amerikanen mogen blij zijn, dat de Japanse 16-megabitchip nog niet rijp is voor produktie op grote schaal. Bij de experimentele chips van nu worden de patronen ingekrast met een heel fijne elektronenbundel. De methode voor massaproduktie werkt via verkleinde projectie van een masker. Dit is een fotografische werkwijze, terwijl de elektronenbundel tekent. Overgaan van tekenen naar fotografie zal naar verwachting pas over enige jaren mogelijk zijn.



Een vergroting van een 1-megabitchip, zoals die op het ogenblik in bepaalde IBM computers wordt toegepast (Foto IBM).

Forse geheugenuitbreiding

Bij de beproefde fotografische produktietechniek treden toch altijd nog vele fabricagefouten op. Men speelt daar helemaal op in door de chip te voorzien van extra cellen, die naar believen de taak kunnen overnemen van slecht uitgevallen condensatortjes. Een 4-megabitchip bevat 96.000 reservecellen. Bedenk daarbij dat een standaard-geheugenchip van een paar jaar geleden 64.000 cellen in totaal had!
Het toewijzen van de reservecellen wordt lastiger naarmate de geheugens groeien. Men krijgt niet alleen last van fouten ontstaan tijdens de fabricage, maar ook van beschadigingen die daarna nog ontstaan. De condensatortjes zijn zo klein dat een toevallig deeltje uit de natuurlijke achtergrondstraling uit de Aarde of uit het heelal het al kan wegvagen. Bij de grotere condensatortjes van vroeger was een dergelijke beschadiging nog niet fataal.

Steeds sneller

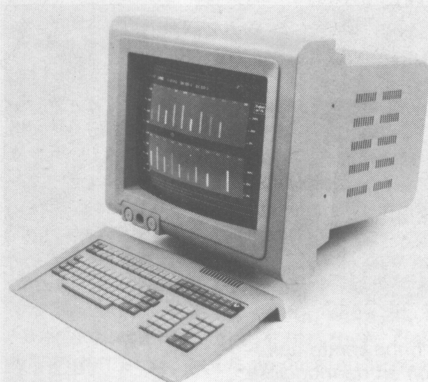
Er moeten daarom extra schakelingen in de chip worden ingebouwd die het opslaan en uitlezen van bits doorlopend controleren. Deze schakelingen beslaan tien procent van het totale oppervlak en maken de chip trager. De controle vergt 5 nanoseconden. (Er gaan een miljard nanoseconden in een seconde.) Eerdere versies van de controleschakelingen vroegen nog 20 nanoseconden. Inclusief controle ligt de toegangstijd in een grote geheugenchip nu rond de 80 nanoseconden. Dat komt neer op het lezen of schrijven van 200 bladzijden in een kwart seconde. Van de 1-megabitgeheugens probeert men inmiddels ook snellere versies te maken. Het record ligt daar op het ogenblik bij 35 nanoseconden.
Bij alle verscheidenheid die in de computerindustrie bestaat, was er altijd een ding waar iedereen zich aan hield: een één werd weergegeven door een elektrische spanning van +5 Volt, een nul door nul Volt. Bij de megabitchips heeft men dit moeten laten vallen. Spanningsverschillen van vijf Volt zouden teveel beïnvloeding geven tussen naburige cellen en signaallijnen. Men is daarom teruggegaan tot 3,3 Volt.

Dr. W. van Tend
Siso code 336.2

Industriële automatisering

Met de hand doen we de afwas in een teiltje sop met behulp van een afwasborstel. We zouden ons een afwasrobot kunnen voorstellen, die op dezelfde manier tewerk gaat. Een afwasmachine gebruikt echter een heel ander principe. Het afwassen is opgesplitst in een aantal opeenvolgende processen. Elk voor zich zijn die geautomatiseerde processen veel eenvoudiger dan ons gedoe met de afwasborstel.

In de industrie zien we op tal van plaatsen hetzelfde verschil als tussen handafwas en machine-afwas. Vroeger werd papier met een zeef geschept uit een bak met papierbrij. In het Openluchtmuseum in Arnhem kunnen we nog zien, hoe dat gaat. Een moderne papiermachine is geen robot die velletjes papier schept, maar een procesmachine, waar een papierbaan doorheenschiet met een snelheid van 25 kilometer per uur. De robot mag dan zeer tot de verbeelding spreken, bij de verdere automatisering van de industrie gaat het ook om procesmachines en om alles wat daartussenin ligt. In Nederland is de belangstelling voor dit onderwerp niet wat die zou moeten zijn. Bij een beurs in Duitsland over automatisering in de landbouw ontstonden kilometers-lange opstoppingen. Toen half maart in de RAI de beurs Industriële Automatisering plaatsvond, leek het alsof we op de verkeerde dag waren gekomen, zo stil was het.



De toetsenborden van ASEA-apparatuur zijn buitengewoon mensvriendelijk: ze zijn bestand tegen gemorste koffie.

Computer ondersteunde produktie

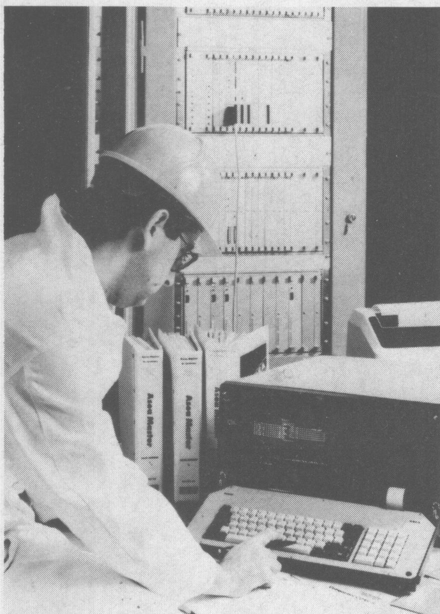
Wie nu spijt heeft van zijn wegblijven, zou zich in verbinding kunnen stellen met het Zweedse bedrijf ASEA, dat in Nederland vestigingen heeft in Apeldoorn en in Capelle aan de IJssel. Daar heeft men verstand van alle aspecten van computerondersteunde produktie, dus niet alleen van robots.
In de procesindustrie moeten vaak allerlei grootheden in de gaten worden gehouden: de toevoer van verschillende grondstoffen, de temperatuur, de druk in een ketel enzovoorts. De computer mag en kan nooit helemaal de baas zijn over onze produktie en dus moeten de grootheden voor de bedieningsman worden weergegeven op beeldschermen. Het gemakkelijkst gaat dat via tekeningen. ASEA heeft daarvoor het Tesselator-systeem bedacht. Letterlijk betekent Tesselator mozaïekmaker. Het beeld wordt hierbij opgebouwd uit kant-en-klare blokjes. Een blokje kan een zelfgemaakt symbool bevatten. In het geheugen bij het beeldscherm ligt opgeslagen, bij welk blokjesnummer welk symbool hoort. De meet- en regelcomputer hoeft alleen het blokjesnummer en de gewenste plaats op het scherm naar het beeldschermstation te sturen en dat station weet er verder raad mee. Bij andere systemen moet de computer telkens de hele opbouw van het symbool doorgeven en dat is natuurlijk veel omslachtiger.
De Tesselatorschermen zijn enkel het vriendelijke gezicht van de fabriek naar de bedieningsman toe. Voor het inwendige van de fabriek heeft ASEA het mastersysteem. Daarmee kan allerlei apparatuur gekoppeld worden. De meet- en regelsystemen van diverse apparaten kunnen dan met elkaar praten en hun activiteiten op elkaar afstemmen.

Koudwals

Een Nederlandse industrie waar met ASEA-apparatuur wordt gewerkt, is Hoogovens IJmuiden. Daar wordt met 'Koudwals 21' staal van 6,25 millimeter dik uitgewalst tot een dikte van 0,3 millimeter. Het mastersysteem van Koudwals 21 staat onder supervisie van minicomputers van het merk DEC. Daarin wordt inge-

Vervolg op blz. 537.

ASEA master is een systeem om allerlei installaties in een fabriek gegevens en commando's te laten uitwisselen.



Op grote beeldschermen kunnen bedieningsmenden precies bijhouden, wat zich in een hele fabriek afspeelt.



Dr. W. van Tend
Siso code 551

Twijfel aan bruine dwerg

Sterren geven zelf licht, planeten weerkaatsen enkel opvallend licht. Bruine dwergen zijn de grensgevallen: ze hebben meer massa dan een planeet, maar te weinig om zelf te kunnen ontbranden. Sinds 1984 dacht men een bruine dwerg te hebben gevonden, namelijk de begeleider van de ster Van Biesbroeck 8. Nu lijkt het erop dat die ster toch geen begeleider heeft.

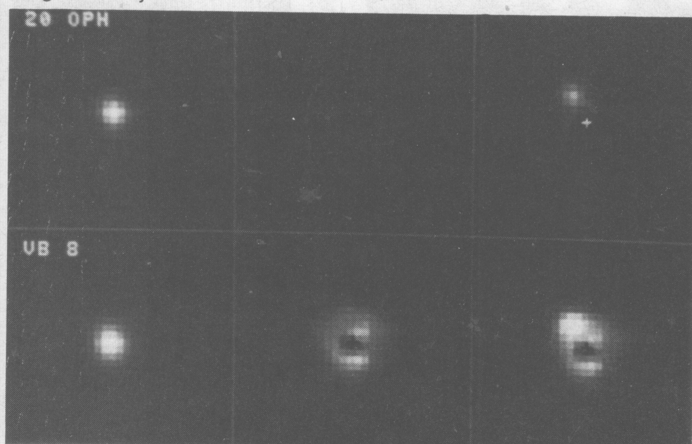
Zware sterren zijn vrij zeldzaam. Gewone sterren als de Zon zijn vrij talrijk. Nòg lichtere sterren zijn er nog meer. Die lijn doortrekkend zouden we denken dat er wel heel veel bruine dwergen zouden moeten zijn. Doordat ze echter maar heel weinig licht uitzenden, zijn ze uiterst moeilijk te vinden. Welke waarneemtechniek men ook neemt, steeds blijkt het zien van bruine dwergen aan de grens van de mogelijkheden te liggen.

Ster nummer 8 uit de catalogus van Van

```
998 rem  Hiervoor de Basicode-2
999 rem  subroutines!
1000 A=500:GOTO20:REM ophiuchus
1010 GOSUB100:REM  scherm schoon
1020 READVE:REM  regelnummer
1030 IFVE=99 THEN1100:REM  klaar
1040 READHO:REM  positie
1050 READTX$:REM  teken(s)
1060 GOSUB110:REM  schermpositie
1070 PRINTTX$;:REM  markeer
1080 V1=VE:H1=HO:REM  onthoud laatste
1090 GOTO1020:REM  volgende ster
1100 HO=3:VE=0:REM bovenaan
1110 GOSUB110:REM  scherm
1120 PRINT"sterrenbeeld Ophiuchus"
1130 VE=1:GOSUB110:REM regel verder
1140 PRINT"(Slangendrager)"
1150 VE=V1:HO=H1:REM  knipperster
1160 GOSUB110:REM  cursor
1170 GOSUB210:REM  wacht op toets
1260 STOP:REM  GOTO950
25000 DATA17,25,"20"
25010 DATA5,26,"o"
25020 DATA6,25,"o"
25030 DATA9,31,"o"
25040 DATA10,15,"*"
25050 DATA11,10,"o o"
25060 DATA11,14,"*"
25070 DATA12,36,"*"
25080 DATA13,34,"*"
25090 DATA17,30,"*"
25100 DATA18,14,"o"
25110 DATA20,22,"*"
25120 DATA5,17,"*"
25130 DATA99
30000 REM
```

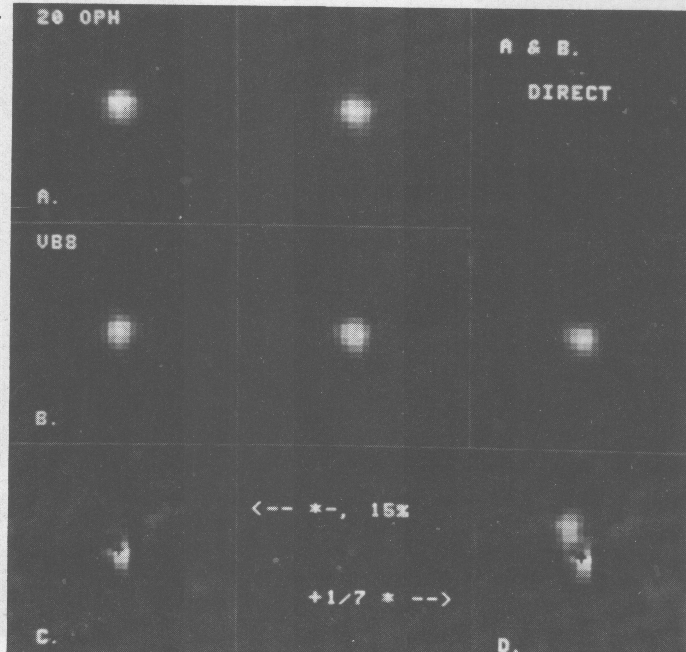
Het computerprogramma waarmee de tekening van Ophiuchus gemaakt is.

Links de CCD-opnamen van 20 Ophiuchi van Van Biesbroek 8. In het midden is een echte puntbron (20 Oph) met aangepaste sterkte afgetrokken van de waarnemingen. Bij vB8 blijft een nevel rond de ster over. Rechts is aan het overblijfsel een kunstmatige begeleider toegevoegd met de eigenschappen van de verwachte vB8B. De kunstmatige begeleider maakt een behoorlijk verschil. Dat geeft aan dat in de middelste afbeelding geen echte begeleider verborgen kan zijn.



Foto's Cornell University.

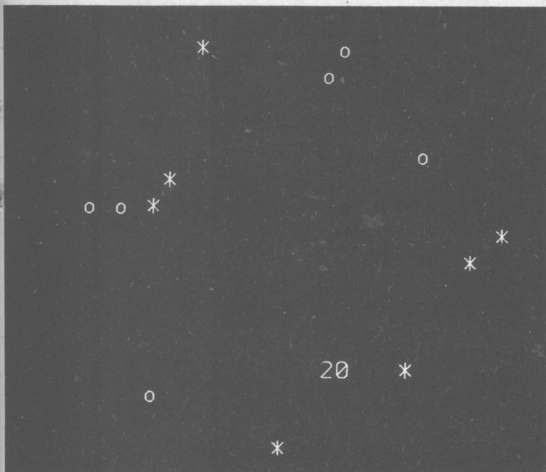
De bovenste twee rijen zijn onbewerkte opnamen van de puntster 20 Oph en de mogelijke dubbelster vB8. Bij C is een puntbron weggehaald uit het plaatje van vB8. In D is een kunstmatige begeleider toegevoegd. De conclusie is dezelfde als bij de andere fotoserie.



Biesbroeck is zonder meer een van de zwakste sterren in onze omgeving. Niettemin is vB8 nog geen bruine dwerg. In 1984 bleek dat vB8 niet alleen was. Wat kon de nog zwakkere begeleider anders zijn dan de eerste echte bruine dwerg? Automatisch kreeg de begeleider de naam vB8B. Het is belangrijk te weten dat vB8B niet zomaar met een kijker gezien was. Men had de lichtgolven van vB8 heel nauwkeurig opgemeten. De conclusie was dat niet alle licht uit hetzelfde punt kwam. Er moest dus nog een tweede, zwakkere lichtbron in de buurt zijn. In feite geen lichtbron maar een infraroodbron, want de hele meting werd gedaan met infrarode straling. De afstand tot de hoofdster moet 1 boogseconde bedragen.
Op de hele wereld is maar één instrument dat infraroodgolven op deze manier kan opmeten. Niemand kon dus de ontdekking precies controleren. Andere mensen moesten andere middelen gebruiken. De Amerikaanse astronomen Skrutskie, Forrest en Shure richtten de infraroodtelescoop van de NASA op Hawaii op vB8. Deze teleskoop heeft een spiegelmiddellijn van drie meter. Twee sterren zo dicht bij elkaar zijn alleen bij een korte belichtingstijd uit elkaar te halen. Anders versmeert de onrust van de lucht het beeld teveel. Men wil echter toch zo lang mogelijk belichten: bij te korte belichting wordt de zwakste ster niet geregistreerd. Het beste compromis was vier opnamen met een belichtingstijd van vijf seconden elk over elkaar heen te leggen. Men gebruikte geen fotografische plaat, maar een CCD-camera, een verfijnde vorm van videoregistratie. De beeldpunten lagen hierbij 0,4 boogseconde uit elkaar. Om een duidelijk plaatje te krijgen, wordt ieder beeldpunt uiteindelijk weergegeven als een blokje. Wanneer de versmering niet al te groot is, moeten twee sterren op een afstand van 1 boogseconde inderdaad uit elkaar te halen zijn. De grootte van de versmering stelt men vast door een vergelijkbare ster in de buurt waar te nemen, in dit geval 20 Ophiuchi. In feite is dat ook een dubbelster, maar de twee sterren staan minder dan 0,05 boogseconde uit elkaar. Dat ze met zijn tweeën zijn, heeft men alleen kunnen afleiden uit de kleursamenstelling van het licht.
Men heeft de opnamen van 20 Oph en vB8 met elkaar vergeleken. Verder is gekeken hoeveel verschil een kunstmatige bruine dwerg zou geven. De conclusie: er is geen begeleider, er is geen bruine dwerg. Wat vB8 extra heeft boven 20 Oph is veel meer een soort nevel. De nevel moet bestaan uit stof dat infrarode straling van de ster verstrooit. Zelf zendt de nevel geen infrarode straling uit, want dan had de IRAS satelliet vB8 moeten zien bij bepaalde golflengten en dat is niet gebeurd. De twee genoemde eigenschappen zijn een beetje vreemd voor een nevel, hoewel dit soort nevels toch wel eens vaker wordt gezien. Het kan natuurlijk zijn dat vB8B eerst naarst vB8A stond en nu precies erachter (of ervoor). Men schat de omlooptijd echter op 50 jaar. In de paar jaar die sinds de eerste waarneming verlopen zijn, kan de verplaatsing nooit voldoende groot zijn geweest. We zitten nu dus weer helemaal zonder bruine dwerg. Op allerlei plaatsen in het heelal moet meer massa aanwezig zijn dan we zien, zo blijkt uit de bewegingen van sterren. De onzichtbare massa zou verborgen kunnen zijn in de vorm van bruine dwergen. Als er echt geen bruine dwergen bestaan, dan moeten zwarte gaten verantwoordelijk worden gesteld voor de ontbrekende massa.
Sommige mensen hebben liever de vrij gewone bruine dwergen dan de exotische zwarte gaten. Bij anderen ligt de voorkeur net anders. Zij zullen het niet bestaan van vB8B aangrijpen om hun stokpaard te ondersteunen. Voor nuchtere astronomen is de strijd nog volkomen onbeslist. Dat vB8B niet te zien is, betekent niet dat er helemaal geen bruine dwergen bestaan. Ze weten zich alleen nog wat onopvallender te gedragen dan wij zouden willen.

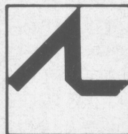
Over bruine dwergen is eerder geschreven in Aarde en Kosmos 5/1986, bladzijde 437.

Van Biesbroeck 8 staat in het sterrenbeeld Ophiuchus in de buurt van het sterretje 20. Dat sterretje is van de vierde magnitude. Van Biesbroeck 8 (A) is in het infrarood vijf magnituden zwakker; de eventuele bruine-dwerg-begeleider nog eens drie magnituden meer. Op zomeravonden staat Ophiuchus (de Slangendrager) aan de zuidhemel.



Wist u dat...

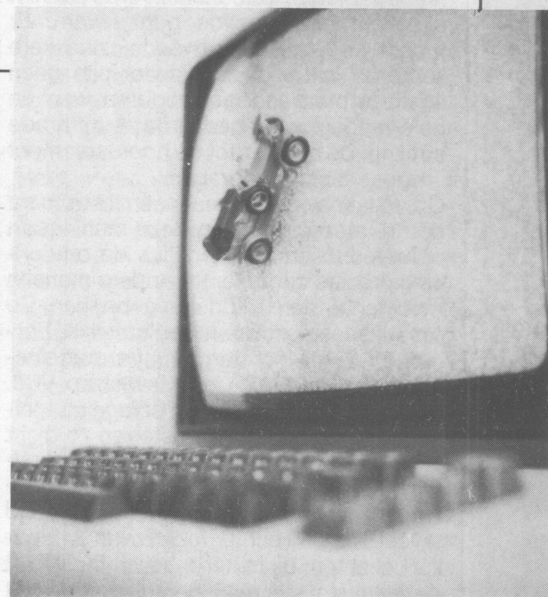
- We na acht jaren weer uit een economisch dal gekomen zijn, maar dat het consumentenvertrouwen nog niet erg groot is.
- Commodore haar prijzen voor de PC/AT, PC-10 en PC-20 sterk verlaagd heeft.
- Kaypro een nieuwe laserprinter op de markt brengt, dat een druksnelheid van acht pagina's A4 per minuut mogelijk is en dat men hiervoor dan meer dan 8000 gulden moet aftikken.
- Philips nieuwe helderrode luminiserende LED's (golflengte 650 nm) op de markt brengt met een hoge lichtintensiteit (tot 200 mcd).
- Op een IBM s36 systeem gemiddeld 1,6 mikrocomputers zijn aangesloten en dat dit gemiddelde op een s38 systeem 4,1 is.
- Japan voorlopig niet meedoet aan de invoering van een BTW van 5% op alle producten.



Ogé Kruijt
Siso code 65

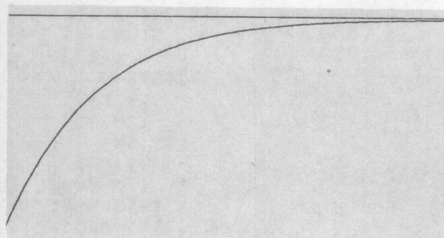
Een autosimulator

Met een eenvoudig computerprogramma kunnen we een automatische snelheidsregeling voor een auto imiteren. Via het programma kunnen we experimenteren met een wagentje dat over een weg rijdt.

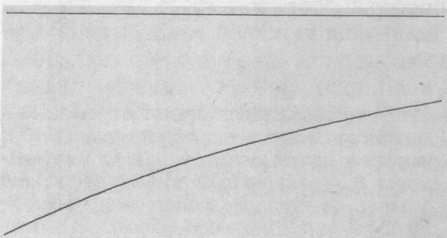


De uitvoer van het programma is een grafiek van de snelheid tegen de tijd. In het begin bevinden we ons in de linkeronderhoek van de grafiek. De tijd T langs de horizontale as is nul, evenals de snelheid V, die langs de verticale as wordt uitgezet. Als de tijd begint te lopen, tekenen we boven punten meer naar rechts steeds de snelheid van het moment. Wanneer de snelheid langere tijd constant blijft, ontstaat een horizontale lijn. Een constante snelheid is wat we gaan proberen te bereiken. Het middel daartoe is de kracht F die de motor via de wielen op de weg uitoefent. Met die kracht duwt de auto de weg naar achteren, wat erop neerkomt dat het voertuig wordt aangedreven. De aandrijfkracht moet twee dingen doen: de auto op gang brengen en de wrijving overwinnen, wanneer de auto op gang is.
Voor de wrijving W geldt de formule die staat in regel 1320 van het programma autosimulator 1. De wrijving neemt toe bij grotere snelheid. Verder is de wrijving groter voor auto's met een grotere massa M. In de formule komt ook nog een constante C voor. Die constante is met name afhankelijk van hoe hard de banden zijn opgepompt. Hoe harder de banden, des te kleiner C.

De formule in regel 1320 gaat over de zogeheten rolwrijving. Er is ook luchtweerstand. Bij grotere snelheden is dat de overheersende weerstandskracht. Ook in de formule daarvoor komt een constante voor, de bekende CW-waarde, die kleiner is bij een betere stroomlijn. Luchtweerstand is niet opgenomen in ons programma. Daarom is het eigenlijk alleen maar geschikt voor lage snelheden. Omdat we met name kijken naar het optrekken van de auto, is dat niet zo erg.
De kracht beschikbaar voor het vergroten van de snelheid is de aandrijfkracht min de wrijving (regel 1370). De verandering DV van de snelheid wordt gegeven door de formule in regel 1380, een van de bewegingswetten van Newton. Hoe groter de massa M van auto plus lading, des te kleiner is de snelheidsverandering. We kijken steeds naar de nieuwe snelheid na een momentje DT. Hoe groter die tijdstap, des te meer de snelheid per stapje verandert. In het eerste programma, autosimulator 1, laten we de auto aandrijven door een constante kracht (regel 1360). Die kracht zal de auto eerst versnellen. Door de toenemende snelheid neemt de wrijving toe (regel 1320). Uiteindelijk wordt de wrijvingskracht gelijk aan de aandrijfkracht. Er is dan geen versnelling meer en de snelheid blijft verder constant.
Een echt regelsysteem is dit niet. Daarvoor zouden we de werkelijke snelheid moeten vergelijken met de snelheid die



Tekening 1.
Grafiek van de snelheid van de auto tegen de tijd. De aandrijving gebeurt door een constante kracht. De grafiek is vijftig tijdseenheden breed (autosimulator 1).



Tekening 2.
De breedte van deze grafiek en van alle volgende is tien tijdseenheden. Bij een constante aandrijfkracht is de auto na die tijd nog maar nauwelijks op gang (aangepaste autosimulator 1).

Listing 1.
Het programma autosimulator 1. De Basicode-3 constanten HG en VG bevatten het aantal beeldpunten in horizontale respectievelijk verticale richting.

```
995 REM  Hiervoor de subroutines
996 REM  van Basicode-3 toevoegen.
997 REM
1000 A=500:GOTO20:REM  autosimulator 1
1010 GOSUB100:REM  scherm schoon
1020 PRINT"Dit programma tekent de"
1030 PRINT"grafiek van het snelheidsverloop"
1040 PRINT"van een voertuig dat"
1050 PRINT"door een constante kracht"
1055 PRINT"wordt voortbewogen."
1060 PRINT
1070 PRINT"Toets voor start>";
1080 GOSUB210:REM  wacht op toets
1200 M=10:REM  massa
1210 C=.1:REM  rolwrijving
1230 VS=6:REM  streefsnelheid
1240 V=0:REM  beginsnelheid
1250 TT=50:REM  eindtijd
1260 DT=.5:REM  tijdsstap
1262 F0=6:REM  constante kracht
1265 VM=VS*1.5:REM  maximaal verwachte snelheid
1270 GOSUB600:REM  grafisch scherm
1280 CN=0:REM  voorgrondkleur
1282 VE=1-(VS/VM):REM  limiet
1284 HO=0:GOSUB620:REM  beginpunt
1286 HO=1-1/HG:GOSUB630:REM  horizontale lijn
1300 FORT=0 TOTT STEPDT:REM  tijd loopt
1320 W=C*M*V:REM  wrijvingskracht
1360 F=F0:REM  constante kracht
1370 FF=F-W:REM  versnellingsdeel
1380 DV=FF/M*DT:REM  snelheidsverandering
1390 V=V+DV:REM  nieuwe snelheid
1500 HO=T/TT*(1-1/HG):REM  plotwaarde tijd
1510 VE=1-V/VM:REM  plotwaarde snelheid
1520 IFT=0 THENGOSUB620:REM  beginpunt
1525 GOSUB630:REM  trek lijntje
1530 NEXTT
1540 GOTO950:REM  stop
32000 REM
32010 REM  Basicode-3
```

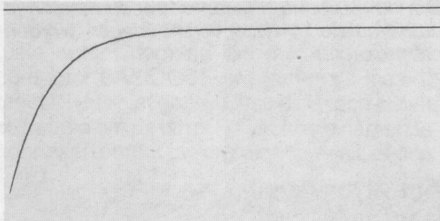
we willen hebben. Op grond van het verschil moeten we de kracht aanpassen. Bij de constante kracht stellen we pas na verloop van tijd vast, op welke snelheid die kracht ons heeft laten uitkomen. De breedte van de grafiek (TT) is 50 tijdseenheden (regel 1250). Binnen die tijd bereikt onze auto inderdaad zijn eindsnelheid. We willen echter dat hij dat al binnen 10 tijdseenheden doet. We passen ook de tijdstap (DT) aan (regel 1260). Geef TT de waarde 10 en DT de waarde 0.05. We tekenen nu weer een grafiek. De constante kracht blijkt in 10 tijdseenheden niet zo erg veel snelheid te halen.

Echte regeling

We gaan nu over op een echte regeling, die we onderbrengen in het programma autosimulator 2. De snelheid die we willen bereiken, noemen we VS (regel 1230), de streefsnelheid. Als we zelf de auto zouden besturen, zouden we aan het begin van de rit veel gas geven (resultaat: een grote kracht). Wanneer we de streefsnelheid zouden naderen, zouden we gas terugnemen. Kortom: we verkleinen de kracht, wanneer het verschil van streefsnelheid en werkelijke snelheid klein wordt.
Dat verschil rekenen we uit bij iedere tijdstap (RP in regel 1330). We passen vervolgens de aandrijfkracht aan aan de eisen van het moment (regel 1360). De constante KP bepaalt, hoe sterk onze motor reageert op het snelheidsverschil. Deze KP heeft vooraf een waarde gekregen (regel 1220).
De snelheid V is nu het resultaat van een wisselende kracht F. Het resultaat V bepaalt op zijn beurt de kracht F bij de volgende tijdstap. Een dergelijke procedure heet terugkoppeling.
Het programma autosimulator 2 geeft in de grafiek de streefsnelheid aan met een horizontale lijn (regels 1282-1286). De werkelijke snelheid komt al heel gauw in de buurt van de streefsnelheid. Maar daarna klopt er toch iets niet. De grafiek van de werkelijke snelheid gaat steeds vlakker lopen. Hij nadert tot een horizontale lijn. Die horizontale lijn ligt echter iets onder onze streefsnelheid. We gaan wel naar een eindsnelheid toe, maar die is kleiner dan onze streefsnelheid. De regeling is niet zuiver. Dit alles wordt duidelijker, wanneer we eens een kleinere waarde van KP proberen (regel 1220). Alleen dankzij de grote KP van 20 kwamen we nog zo dicht bij de streefsnelheid. Een grote KP betekent dat onze motor heel fel reageert op een snelheidsverschil. Willen we met een minder felle motor de streefsnelheid halen, dan is een slimmer regelsysteem nodig.
We zijn een beetje gehandicapt, doordat onze auto alleen een snelheidsmeter heeft. Dat we de streefsnelheid niet echt haalden, kwam door de wrijving. De regeling onderneemt pogingen op grond van het snelheidsverschil, maar de wrijving knaagt het resultaat steeds weer aan. We zouden een wrijvingsmeter moeten hebben om rekening te houden met dat aanknagen. Wrijvingsmeters zitten echter niet in auto's. We moeten het echt doen met alleen een snelheidsmeter.

Geheugen erbij

Autosimulator 2 gebruikte de stand van de snelheidsmeter op dit moment. In autosimulator 3 gaan we ook de standen op vorige momenten verwerken. We geven de regeling een geheugen. De snelheid bleef eerst hangen onder de streefsnelheid. Wanneer we al die verschillen onthouden, wordt de ernst van het verschil steeds groter. Dat veroorzaakt een steeds grotere bijregeling, die op een gegeven moment het aanknagen door de wrijving overtreft. Zo worden we de wrijving de baas zonder die te hoeven meten.
In autosimulator 3 tellen we alle snelheidsverschillen sinds het begin bij elkaar in VI (regel 1345). VI moet wel op nul beginnen (regel 1295). Er moet een constante KI worden vastgelegd, die de felheid van de reactie van de motor aangeeft (regel 1222). KI wordt gebruikt in regel 1350. Regel 1360 berekent de kracht, waarbij overigens nog steeds ook de oude bijregeling op het verschil van het moment blijft werken. In de grafiek zien we dat nu de streefsnelheid wel wordt bereikt. Doordat een vroeger verschil niet wordt vergeten, wordt de streefsnelheid zelfs overschreden. Die overschreiding zorgt ervoor dat in regel 1345 negatieve getallen bij VI opgeteld gaan worden. Op den duur wordt VI daardoor nul en komen we zuiver op de streefsnelheid uit.
We hebben al gezegd dat de wrijvingsconstante C (regel 1210) afhangt van hoe hard de banden zijn opgepompt. We gaan nu een klapband nabootsen door na 5 tijdseenheden C te vergroten. Daarvoor voegen we een regel 1310 toe. Zo ontstaat het programma autosimulator 3+, waarin tweemaal wordt geprobeerd de streefsnelheid te bereiken. Daarbij gaan we er wel van uit dat na de klapband de aandrijving normaal blijft werken. Het moet dus een niet aangedreven wiel zijn, waarvan de band klapt.
We doen inderdaad steeds alsof we de auto kunnen aandrijven met een kracht naar wens. In de reclame voor auto's staat echter niet een aandrijfkracht, maar een motorvermogen in pk's of - moderner - in kilowatt. Tussen de motor en het wegdek zit nogal het een en ander, bijvoorbeeld een versnellingsbak. Hoeveel van het motorvermogen wordt gebruikt als aandrijfvermogen is daarom niet zo eenvoudig te zeggen. In de praktijk zal het niet zo eenvoudig zijn de motor zo te regelen dat via de tussenstappen een kracht naar wens tot stand komt. Als ons dat toch lukt, en als er geen verliezen zijn, geldt dat het vermogen gelijk is aan de kracht maal de snelheid. Met het programma autosimulator 4 wordt de grafiek van het vermogen getekend - in plaats van de grafiek van de snelheid. Voor de rest heeft dit programma dezelfde werking als autosimulator 3+ (dus met geheugenregeling en klapband).
In autosimulator 1 hadden we een constante kracht en een langzaam toenemen-

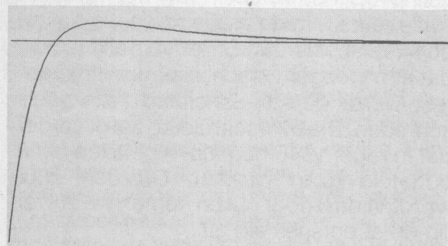


Tekening 3.
De snelheid bij regeling op de verschilsnelheid. Er is een eindsnelheid, die onder de streefsnelheid ligt (autosimulator 2 met KP=10).

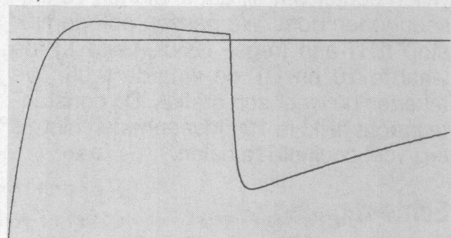
Listing 2.
Het computerprogramma autosimulator 2.

```
995 REM  Hiervoor de subroutines
996 REM  van Basicode-3 toevoegen.
997 REM
1000 A=500:GOTO20:REM  autosimulator 2
1010 GOSUB100:REM  scherm schoon
1020 PRINT"Dit programma tekent de"
1030 PRINT"grafiek van het snelheidsverloop"
1040 PRINT"van een voertuig dat"
1050 PRINT"een streefsnelheid probeert"
1055 PRINT"te halen."
1060 PRINT
1070 PRINT"Toets voor start>";
1080 GOSUB210:REM  wacht op toets
1200 M=10:REM  massa
1210 C=.1:REM  rolwrijving
1220 KP=20:REM  regelsterkte
1230 VS=6:REM  streefsnelheid
1240 V=0:REM  beginsnelheid
1250 TT=10:REM  eindtijd (kleiner)
1260 DT=.05:REM  tijdsstap (kleiner)
1262 F0=6:REM  standaardkracht (zonder functie)
1265 VM=VS*1.5:REM  maximaal verwachte snelheid
1270 GOSUB600:REM  grafisch scherm
1280 CN=0:REM  voorgrondkleur
1282 VE=1-(VS/VM):REM  limiet
1284 HO=0:GOSUB620:REM  beginpunt
1286 HO=1-1/HG:GOSUB630:REM  horizontale lijn
1300 FORT=0 TOTT STEPDT:REM  tijd loopt
1320 W=C*M*V:REM  wrijvingskracht
1330 RP=VS-V:REM  discrepantie
1360 F=KP*RP:REM  bijregeling
1370 FF=F-W:REM  versnellingsdeel
1380 DV=FF/M*DT:REM  snelheidsverandering
1390 V=V+DV:REM  nieuwe snelheid
1500 HO=T/TT*(1-1/HG):REM  plotwaarde tijd
1510 VE=1-V/VM:REM  plotwaarde snelheid
1520 IFT=0 THENGOSUB620:REM  beginpunt
1525 GOSUB630:REM  trek lijntje
1530 NEXTT
1540 GOTO950:REM  stop
```

de snelheid. Dat betekent dat daar het vermogen ook langzaam toenam tot een eindwaarde. Bij een kracht FO en een eindsnelheid VS is het eindvermogen FO maal VS. Dat is ook het vermogen dat in autosimulator 3+ en 4 wordt ontwikkeld vlak voordat de band klapt. We zien in de grafiek dat dat vermogen nogal onderin ligt. Tijdens het optrekken was het vermogen vele malen hoger. Het snellere optrekken in de programma's autosimulator 2, 3 en 4 is dus te danken aan een tijdelijk hoog vermogen. Daarvoor hebben we een sterkere motor nodig. Veel vermogen ontwikkelen betekent ook veel brandstof gebruiken. Bij rijden in de stad moeten we



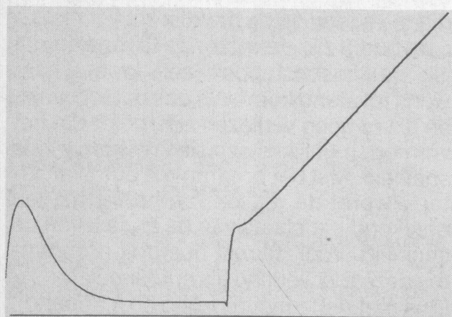
Tekening 4.
Wanneer rekening wordt gehouden met afwijkingen over langere tijd, kan de eindsnelheid gelijk worden aan de streefsnelheid (autosimulator 3).



Tekening 5.
Na een klapband probeert de regeling opnieuw de streefsnelheid te bereiken (autosimulator 3+).

Listing 3.
Het programma autosimulator 3.

```
995 REM  Hiervoor de subroutines
996 REM  van Basicode-3 toevoegen.
997 REM
1000 A=500:GOTO20:REM  autosimulator 3
1010 GOSUB100:REM  scherm schoon
1020 PRINT"Dit programma tekent de"
1030 PRINT"grafiek van het snelheidsverloop"
1040 PRINT"van een voertuig dat"
1050 PRINT"een streefsnelheid probeert"
1055 PRINT"te halen. Met geheugen."
1060 PRINT
1070 PRINT"Toets voor start>";
1080 GOSUB210:REM  wacht op toets
1200 M=10:REM  massa
1210 C=.1:REM  rolwrijving
1220 KP=20:REM  regelsterkte
1222 KI=.02:REM  sterkte geheugenregeling
1230 VS=6:REM  streefsnelheid
1240 V=0:REM  beginsnelheid
1250 TT=10:REM  eindtijd (kleiner)
1260 DT=.05:REM  tijdsstap (kleiner)
1262 F0=6:REM  standaardkracht (zonder functie)
1265 VM=VS*1.5:REM  maximaal verwachte snelheid
1270 GOSUB600:REM  grafisch scherm
1280 CN=0:REM  voorgrondkleur
1282 VE=1-(VS/VM):REM  limiet
1284 HO=0:GOSUB620:REM  beginpunt
1286 HO=1-1/HG:GOSUB630:REM  horizontale lijn
1295 VI=0:REM  snelheidsgeheugen schoon
1300 FORT=0 TOTT STEPDT:REM  tijd loopt
1320 W=C*M*V:REM  wrijvingskracht
1330 RP=VS-V:REM  discrepantie
1345 VI=VI+(VS-V):REM  tel verschil bij
1350 RI=VI*KI:REM  geheugenkracht/KP
1360 F=KP*(RP+RI):REM  bijregeling
1370 FF=F-W:REM  versnellingsdeel
1380 DV=FF/M*DT:REM  snelheidsverandering
1390 V=V+DV:REM  nieuwe snelheid
1500 HO=T/TT*(1-1/HG):REM  plotwaarde tijd
1510 VE=1-V/VM:REM  plotwaarde snelheid
1520 IFT=0 THENGOSUB620:REM  beginpunt
1525 GOSUB630:REM  trek lijntje
1530 NEXTT
1540 GOTO950:REM  stop
```



Tekening 6.
Het optrekken van de auto veroorzaakt een piek in het benodigde motorvermogen. De terugval van de snelheid bij de klapband laat heel even ook het vermogen teruggaan. Onmiddellijk daarna begint de regeling heel veel vermogen van de motor te vragen (autosimulator 4).

```
1000 A=500:GOTO 20:REM  autosimulator 3+

1055 PRINT "te halen. Met geheugen en"
1057 PRINT "klapband."

1310 IF T>5 THEN C=5:REM  klapband
```

▲ *Listing 4.*
Om de klapband te laten optreden moeten in autosimulator 3 deze veranderingen worden aangebracht.

Listing 5.
Het programma autosimulator 4.

```
995 REM  Hiervoor de subroutines
996 REM  van Basicode-3 toevoegen.
997 REM
1000 A=500:GOTO20:REM  autosimulator 4
1010 GOSUB100:REM  scherm schoon
1020 PRINT"Dit programma tekent de"
1030 PRINT"grafiek van het vermogensverloop"
1040 PRINT"van een voertuig dat"
1050 PRINT"een streefsnelheid probeert"
1055 PRINT"te halen. Met geheugen en"
1057 PRINT"klapband."
1060 PRINT
1070 PRINT"Toets voor start>";
1080 GOSUB210:REM  wacht op toets
1200 M=10:REM  massa
1210 C=.1:REM  rolwrijving
1220 KP=20:REM  regelsterkte
1222 KI=.02:REM  sterkte geheugenregeling
1230 VS=6:REM  streefsnelheid
1240 V=0:REM  beginsnelheid
1250 TT=10:REM  eindtijd (kleiner)
1260 DT=.05:REM  tijdsstap (kleiner)
1262 F0=6:REM  standaardkracht
1265 VM=VS*F0*20:REM  maximaal verwacht VERMOGEN
1270 GOSUB600:REM  grafisch scherm
1280 CN=0:REM  voorgrondkleur
1282 HO=0:VE=1-1/VG:REM  linksonder
1284 GOSUB620:REM  beginpunt
1286 HO=1-1/HG:REM  rechtsonder
1290 GOSUB630:REM  lijn
1295 VI=0:REM  snelheidsgeheugen schoon
1300 FORT=0 TOTT STEPDT:REM  tijd loopt
1310 IFT>5 THENC=5:REM  klapband
1320 W=C*M*V:REM  wrijvingskracht
1330 RP=VS-V:REM  discrepantie
1345 VI=VI+(VS-V):REM  tel verschil bij
1350 RI=VI*KI:REM  geheugenkracht/KP
1360 F=KP*(RP+RI):REM  bijregeling
1370 FF=F-W:REM  versnellingsdeel
1380 DV=FF/M*DT:REM  snelheidsverandering
1390 V=V+DV:REM  nieuwe snelheid
1500 HO=T/TT*(1-1/HG):REM  plotwaarde tijd
1510 VE=1-V*F/VM:REM  plotwaarde vermogen
1520 IFT=0 THENGOSUB620:REM  beginpunt
1525 GOSUB630:REM  trek lijntje
1530 NEXTT
1540 GOTO950:REM  stop
32000 REM
```

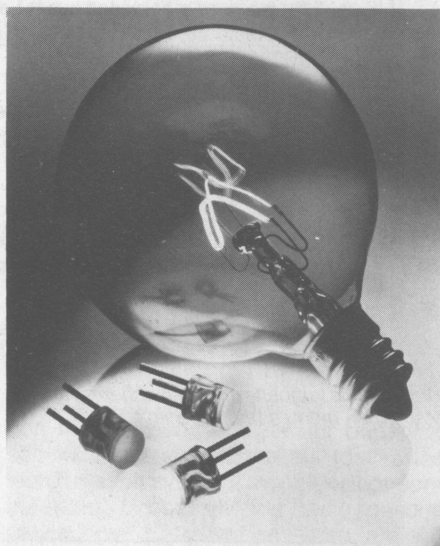
telkens weer opnieuw optrekken met hoog vermogen. Daarom is rijden in de stad naar verhouding duur. De optrekenergie kunnen we in principe terugwinnen bij het afremmen. Gewone auto's doen dat niet. Bij elektrische voertuigen is het wel mogelijk. De motor wordt dan omgeschakeld tot een dynamo, die vervolgens dienst doet als rem. Voor auto's is elektrische aandrijving niet praktisch omdat accu's zo zwaar zijn - het meevoeren van accu's kost ook weer veel energie. Treinen kunnen een bovenleiding gebruiken en daar wordt elektrische terugwinremming al tientallen jaren toegepast. In Zwitserland leveren drie treinen die bergaf gaan, de energie voor één trein bergop. (Bergop rijden vraagt ongeveer hetzelfde als optrekken.)

Door de grote massa van auto plus lading is bij een vrachtauto ook voor het rijden met constante snelheid al een groot vermogen nodig. De motor heeft maar weinig extra vermogen voor het optrekken, wat dientengevolge langer duurt dan bij een personenwagen. Het laten toenemen van de snelheid vraagt bij een vrachtauto veel brandstof. Transportondernemers willen dan ook dat hun chauffeurs zoveel mogelijk met constante snelheid rijden. Tegenwoordig wordt ook met allerlei ingebouwde regelelektronica geprobeerd motoren zo zuinig mogelijk te laten werken. Soms wordt zuinigheid echter ingeleverd voor kracht. Dat is bijvoorbeeld zo bij trekkertrekwedstrijden. Daarbij moet een aanhanger worden voortgetrokken, waarop een mechaniek zit, dat ervoor zorgt dat de wrijvingsconstante C geleidelijk aan groter wordt. Op een gegeven moment loopt de trekker dan vast in het zand, de wielen slippen of de motor gaat in rook op. Wie het verst komt, wint. Ons klapbandprogramma laat de wrijvingsconstante C ineens toenemen, van 0.1 naar 5. Die toename is misschien wel wat groot. De bijregeling van de snelheid gaat zeer zware eisen stellen aan het vermogen. Trekkertrek kunnen we nabootsen door C vanaf een kleine waarde met ieder tijdstapje wat groter te maken. Een waarde voor de kracht waarbij de wielen gaan slippen en een vermogen waarbij de motor het begeeft, kunnen we naar eigen inzicht inbouwen.

Vervolg van blz. 531

voerd wat voor plaat men precies wil hebben en kleinere computers besturen dan de walsinstallaties. Koudwalsen gebeurt bij kamertemperatuur. Er bestaan ook warmbandwalsen. Daar begint men met plakken staal van maar liefst 20 centimeter dikte, 2 meter lang en 90 centimeter breed. Dit soort plakken wordt omgevormd tot rollen dun staal (dikte 1,25 millimeter). Men regelt de installatie met twee procescomputers, die ieder leiding geven aan vier stuurcomputers. Op het ogenblik worden al die computers voorzien van Tesselatorbeeldschermen. Die komen te staan in vijf stuurhuizen. Vandaaruit overziet de bedieningsman de produktie, letterlijk, maar ook op het scherm.
Van het staal dat bij Hoogovens wordt gefabriceerd, kan men onder andere auto's maken. Dat gebeurt in Nederland bij Volvo in Born, Limburg. De vlakke staalplaten moeten daar de vorm krijgen van bijvoorbeeld een portier of een spatbord. Dat gaat met een aantal opeenvolgende bewerkingen in 800- of 1200-tons persen. Van elke soort zijn er zes, die met tussenruimten van vijf meter staan opgesteld in zogenaamde persstraten. Tussen de persen in staan robots. Die pakken de onderdelen-in-wording uit de ene pers en leggen ze neer in de volgende, die het onderdeel weer iets dichter bij de uiteindelijk gewenste vorm brengt. De 800-tons persen werkten zo snel dat de robots ze niet konden bijbenen (of eigenlijk bij-armen). Volvo is daarom samen met ASEA de robots gaan herprogrammeren om ze sneller te maken. In totaal worden bij Volvo in Born 120 robots gebruikt, waarvan 84 van ASEA. Behalve bij het persen zijn ze betrokken bij puntlassen en bij montage.

Gloeilampvervangende LED



Hewlett en Packard introduceert een totaal nieuw LED-lamp-concept. Het betreft hier een lampje met een diameter van 6 mm en een zeer hoge lichtopbrengst. Vier LED-chips zorgen voor een egale verlichting van het oppervlak. Het stroomverbruik is laag (50 mA bij circa 4,5 Volt).
De lamp is verkrijgbaar in 3 kleuren: groen, rood en geel. De firma Diode in Houten importeert deze kleurige schijnsels. (R.G.L.)

Nieuwe portofoon

Communiceren is in deze maatschappij een van de belangrijkste zaken om berichten en gegevens over te brengen. De firma Koning en Hartman te Delft werkt daar ook aan mee door een nieuwe portofoon op de markt te lanceren: de Storno CQP4000 MK2. Het standaard apparaat met oplaatbare accu weegt slechts 600 gram en is spatwater- en stofdicht.
De portofoon werkt in de frequentie-modulatie-band tussen de 146-174 MHz (VHF) of 420-470 MHz (UHF). Het zendvermogen bedraagt 1 of 2 Watt en verder is dit apparaat volledig programmeerbaar. (R.G.L.)



Dr. W. van Tend
Siso code 664.2

Ook Apple computers bij Universiteit Utrecht

Behalve MS-DOS-computers gaat de Rijksuniversiteit Utrecht ook Apple Macintosh-machines verspreiden onder medewerkers en studenten. In het bedrijfsleven zijn de MS-DOS-computers bezig alle andere systemen te verdringen. Apple lijkt nu via de universiteiten een poging te ondernemen nog een klein stukje van de computermarkt te behouden.

Tot nu toe verkocht de computerwinkel van de Utrechtse universiteit bijna uitsluitend MS-DOS-computers van de merken Olivetti en IBM. In Amsterdam is men al wel langer bezig met Macintosh-machines. Bij de twee Amsterdamse universiteiten staat er reeds een duizendtal. Men is zelfs opgehouden met de verkoop van MS-DOS-machines, niet vanwege de computers zelf, maar door problemen met de leveranciers. Uit Amsterdam klinkt nu de kreet: "Een verschil van opvatting tussen een MS-DOS-aanhanger en een Macintosh-gebruiker kan alleen worden voorkomen, wanneer de MS-DOS-aanhanger nog nooit een Macintosh heeft geprobeerd." Laten we dat maar eens wat nuchterder bekijken.

Commando's

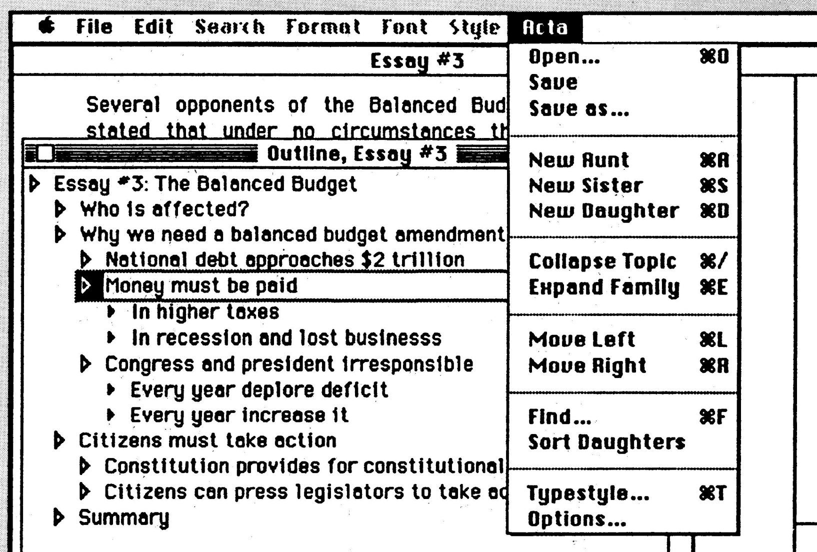
Wanneer iemand een MS-DOS-computer iets wil laten doen, moet hij daarvoor een commando intypen. Het commando:

COPY A:TEKST1.TXT B:TEKST2.TXT

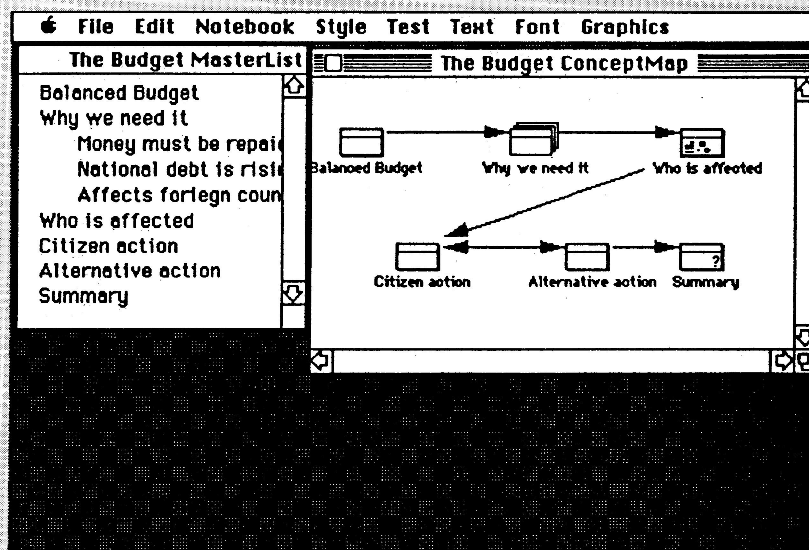
neemt de informatie die op diskette A staat onder de naam TEKST1.TXT, en maakt daarvan een kopie op diskette B. Die kopie krijgt de naam TEKST2.TXT. Voor de komst van MS-DOS hadden minicomputers al vele jaren met dergelijke commando's gewerkt. Wie een minicomputer gewend was en een MS-DOS-microcomputer te zien kreeg, vond dit best mooi. Het werkte allemaal net zo als op de mini, alleen nu op een veel goedkopere machine.

Bij een Macintosh zijn er geen commando's als COPY, die ingetypt moeten worden. Wanneer de computer aanstaat, is op het scherm altijd een aantal symbooltjes of een rijtje trefwoorden te zien. Verder staat op het scherm een pijltje. Naast de computer bevindt zich de zogenaamde muis. Dit is een blokje met aan de onderkant een bolletje. Wanneer we de muis over het buro schuiven, komt het bolletje in beweging. Het rollen van het bolletje wordt vertaald in een verschuiving van het pijltje op het scherm. Zo kunnen we via de muis op het scherm iets aanwijzen.

Op de muis zit een knop. Drukken we die in, dan wordt datgene gekozen wat het pijltje aanwijst. Wijzen we bijvoorbeeld een diskette aan, dan kiezen we met de knop de inhoud van die diskette. In die inhoud op het scherm kunnen we weer iets aanwijzen en kiezen. Dan verschijnt bijvoorbeeld een bepaalde tekst die op de diskette is opgeslagen. Een dergelijke tekst komt altijd in een venster op het scherm te staan. Als het geheel niet in het venster past, dan kunnen we als het ware het document onder het raam doorschuiven, waarbij we steeds nieuwe stukken tekst kunnen bekijken.



Met de muis kunnen we op het scherm van een Macintosh menu's tevoorschijn toveren.



De Macintosh kent geen afzonderlijke tekstschermen en grafische schermen. Tekst en tekeningen kunnen zonder problemen bij elkaar worden gezet.

Dit alles heeft bepaalde opvallende voordelen ten opzichte van het systeem van MS-DOS. Aanwijzen is gemakkelijker dan een naam intypen. We kunnen alleen iets aanwijzen dat ook werkelijk bestaat. Als we een naam verkeerd invoeren, verwij-

zen we naar iets dat niet bestaat en we brengen de computer in de problemen. Hij moet dan een foutmelding geven en wij moeten weer weten wat we daarmee aan moeten.

Dezelfde dingen op een andere manier

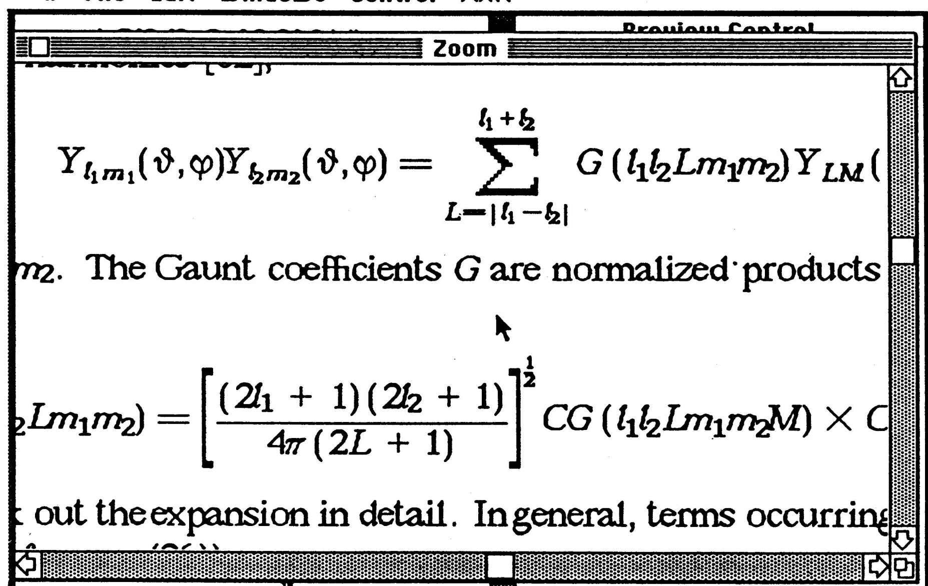
Alle software voor de Macintosh werkt met aanwijzen via de muis volgens precies hetzelfde systeem. Bij MS-DOS is er bepaald geen vast systeem. Willen we rechtstreeks vanuit MS-DOS de naam van een verzameling gegevens veranderen, dan is het commandowoord RENAME. Willen we hetzelfde doen vanuit Basic, dan is het trefwoord NAME. Willen we gegevens op diskette kwijt, dan gaat dat rechtstreeks met ERASE maar vanuit Basic met KILL. Met andere woorden: iedere keer moeten we dezelfde dingen op een andere manier leren doen. Willen we overigens bij de Macintosh gegevens kwijt, dan moeten we een prullebakje aanwijzen.

Van de Macintosh wordt gezegd dat alles zichzelf wijst en dat geen gebruiksaanwijzingen nodig zijn. Helemaal waar is dat niet. Veel dingen zijn aangegeven met plaatjes zoals ook op stations te zien zijn. Een poppetje op een stoel met een koffer ernaast verwijst naar de wachtkamer; een rechthoekje met een pijl eruit naar de uitgang; een sigaret met een kruis erdoor betekent niet roken. Wie op een willekeurig moment naar het scherm van een Macintosh kijkt, zal niet meteen alle symbooltjes, trefwoorden en vensters begrijpen. Uitproberen of in een handleiding kijken zijn dan toch nog de aangewezen middelen.

Wie een Macintosh koopt, krijgt daarbij helemaal geen toepassingsprogramma's en evenmin enige programmeertaal. Met een kale Macintosh valt dus niets te doen; er zal eerst software moeten worden aangeschaft. Macintosh-software is gelukkig in het algemeen wat goedkoper dan vergelijkbare programma's voor MS-DOS. Volgens Apple zijn de kopers van de Macintosh voor een groot deel mensen die niet zelf willen programmeren. Voor die mensen is het dan ook inderdaad gemakkelijker niet te hoeven werken met commando's die toch verband houden met de vaktaal van programmeurs.

Mensen die wel zelf programma's willen schrijven in de aloude programmeertalen, zullen vanuit een wat andere hoek aankijken tegen de Macintosh. De programmeertalen die voor de Macintosh kunnen worden aangeschaft, zijn uitgebreid met extra opdrachten om te kunnen werken met de muis en de menu's op het scherm. Bij uitvoering spreken die opdrachten subroutines aan die vastliggen in het ROM-geheugen van de Macintosh. In de nieuwste versie heeft dat ROM een omvang van maar liefst 256 K. Vanuit Pascal kan dit hele vaste geheugen ten volle worden uitgebuit, vanuit Basic minder. Wie gewend is op een gewone computer te programmeren, zal de omhaal van de Macintosh vaak als hinderlijk ervaren. Het maken van een programma dat werkt via muis, menu's en symbooltjes kost tweemaal zoveel tijd als het ontwikkelen van een gewoon programma.

Het bedieningssysteem van de Macintosh is inmiddels ook nagebootst bij computers van andere merken. In de MS-DOS-wereld werkt het programma Framework met een vergelijkbaar menusysteem. Voor MS-DOS is er tegenwoordig ook MS-WINDOWS, een meer algemeen venstersysteem. Dit werkt overigens alleen goed op snelle MS-DOS-computers met veel geheugen en met een vaste schijf. De grote mikrocomputers van Atari hebben ook hun schermbeeld van de Macintosh afgekeken. Het muis-en-menu's-systeem is daar echter maar een dun schilletje, niet iets waarvan de hele computer doordrenkt is.



Met de wetenschappelijke tekstverwerker TEX kunnen de meest ingewikkelde formules worden opgesteld.

Verschil tussen mikroprocessors

In vele opzichten zijn de Macintosh en MS-DOS onverenigbaar. Evenals de Atari's en de Comodore Amiga heeft de Macintosh een 68000 of 68020 mikroprocessor als zenuwcentrum. Bij MS-DOS-computers is dit een 8088 chip (of een afgeleide daarvan: 8086, 80286, 80386). Dit verschil tussen mikroprocessors die met een zes beginnen en mikroprocessors met tachtig is al heel oud. In feite gaat het hier om twee verschillende chipfabrikanten: de 68000 is een produkt van Motorola, de 80-serie is ontwikkeld door Intel.

Voor de Macintosh bestaan wel uitbreidingskaarten voor het draaien van MS-DOS-programma's. In feite bevat een dergelijke kaart alle elektronica voor een 80xx-computer. Met de kaart ontstaat een soort siamese-tweelingcomputer met alle gevolgen van dien. In het algemeen is een zo sterke band niet nodig. Dan is koppeling via een telefoonlijn voldoende om gegevens uit te wisselen. Bij de Rijksuniversiteit Utrecht gaat men ook op grote schaal dit contact bedrijven. Alle computers moeten zich hierbij aanpassen aan het verbindende netwerk, en Apples kunnen dat net zo goed als MS-DOS-machines.

Zolang het alleen gaat om het uitwisselen van eenvoudige letters en cijfers, kennen vrijwel alle computers dezelfde codes. (Onder de thuiscomputers is de Commodore 64 de beruchte uitzondering.) Wanneer het gaat om letters met accenten of om dingen als het guldensymbool of het wortelteken, dan zijn de codes verschillend voor alle verschillende types. In wetenschappelijke teksten komen nogal wat van die symbolen voor die niet uitwisselbaar zijn. Er is daarom een manier bedacht om zelfs de meest bijzondere tekens te kunnen weergeven in een uitwisselbare code. Dit systeem heet TEX, wat om onduidelijke redenen wordt uitgesproken als "tech". Hierbij zijn er in feite steeds twee teksten. Eén bevat alleen eenvoudige tekens en commando's. Die tekst kan van de ene computer naar de andere worden gestuurd. De andere tekst is de vertaling daarvan tot de ingewikkelde tekens en formules die we eigenlijk bedoelen. Alleen de eerste tekst kunnen we zelf bewerken. Steeds moet een vertaalprogramma ons het resultaat in de tweede vorm tonen. In Amsterdam heeft men voor de Apple-Macintosh zo'n vertaalprogramma gemaakt. Dit systeem van verwerking is te langzaam om aangenaam te zijn, maar die ergernis is er op andere computers ook.

Macintosh creatiever

Met de Macintosh vervreemden de universiteiten zich nogal van het bedrijfsleven, dat alom MS-DOS gebruikt. Omdat het bedieningsgemak van de Macintosh in het onderwijs zo goed van pas komt, is dat wel een beetje verdedigbaar. De Macintosh is vanouds wel op wat grotere schaal toegepast in de creatieve sector: de reclamewereld en de muziekindustrie. Men heeft de Macintosh wel eens de duurste

tekendoos ter wereld genoemd. Bij de Hogeschool voor de Kunsten Utrecht wordt de Mac ingezet als hulp bij het componeren van synthesizermuziek. Als een wat algemenere creatieve toepassing wordt tegenwoordig desktop-publishing genoemd: het samenstellen van tijdschriftpagina's (inclusief foto's, inclusief kleur) op het beeldscherm. Daarbij vervalt een groot aantal stappen van het klassieke produktieproces. Ook voor MS-DOS is desktop-publishing in opkomst. Daarvoor is er echter nog geen programma dat Nederlandse woorden aan het einde van een regel (redelijk) correct afbreekt, voor de Macintosh wel.

Antwoord op brief

De Rijksuniversiteit Utrecht gaat niet alleen Macintosh-computers verspreiden, men is ook toegetreden tot het Apple University Consortium. Dit is een vereniging van universiteiten die hun Macintosh-ervaringen uitwisselen. In Nederland zijn de Amsterdamse universiteiten al eerder lid geworden, evenals die van Nijmegen en Leiden. In totaal bestaat het consortium uit 30 Europese en 24 Amerikaanse universiteiten. Men houdt contact via een tijdschrift 'Wheels for the mind'. Daarin melden de leden welke programma's zij ontwikkeld hebben.
Een van die meldingen mag niet onvermeld blijven. Het gaat om het programma Plate Tectonics, gemaakt door Prof. M. McWilliams van Stanford University, Californië. Met dit programma kan de drift van de continenten worden nagebootst. Het is daarmee precies het antwoord op een brief die we kregen van H. Veenstra uit Groningen. Als deze meneer dus even een Macintosh aanschaft...

Wist u dat...

- Mitsubishi Electric onlangs heeft bekend gemaakt dat een 1-Mbit-EPROM, geproduceerd volgens de CMOS-technologie, aan het leveringsprogramma is toegevoegd.
- Op 8 april j.l. de 20 miljoenste exportauto, de Nissan Sunny, met bestemming Finland vanuit Japan verscheept is en dat deze export in een tijdsbestek van 6 jaar heeft plaatsgevonden.
- De groei in de automatiseringsmarkt zich gestadig met zo'n 20 tot 25% per jaar doorzet.
- Hewlett en Packerd een nieuwe draagbare spectrumanalyser tot 10 KHz heeft (HP 8590A) en dat deze op PC's kan worden aangesloten.
- De tekstverwerker voor PC's, Word Perfect 4.2, weer beter geperfectioneerd is en dat volgens de importeur er een 5.0 versie op komst is die alle andere versies overtreft.

Ogé Kruijt
Siso code 532.5

Kogelbanen

De baan die een afgeschoten kogel beschrijft, is te voorspellen aan de hand van enkele natuurwetten. In het voortgezet onderwijs wordt de berekeningsmethode onderwezen in de natuurkundeles. Meestal gaat men dan uit van eenvoudige omstandigheden. Luchtweerstand wordt bijvoorbeeld meestal niet meegenomen in de berekening.



Met de computer als hulpmiddel wordt de berekening plotseling veel eenvoudiger. We kunnen de kogelbaan meteen op het scherm zien. Verschijnselen als luchtweerstand en tegenwind geven nauwelijks meer werk.
In de berekening nemen we de zwaartekracht op alle hoogten gelijk. In werkelijkheid neemt de zwaartekracht iets af met de hoogte. Bij lage kogelbanen is die afname echter zo gering dat het effect toch niet op het scherm te zien zou zijn. Verder gaan we werken met een wind die zuiver horizontaal waait. Onze kogel krijgt de wind mee of tegen; er is geen zijwind.
We schrijven het programma in Basicode-3, waardoor het zonder al te veel problemen bruikbaar wordt voor computers van verschillende merken. Een dergelijk programma heeft regels 1000 en 1010 met een vaste vorm.

```
1000 A=500:GOTO 20:REM kogelbaan
1010 GOSUB 100:REM scherm schoon
```

Aan het begin van het programma gaan we een aantal gegevens invoeren. Die leggen de situatie van het kanon tijdens het schot vast.De afvuurcondities zijn de beginsnelheid (V) van de kogel en de elevatie (EL) van het kanon. De elevatie is de hoek waarmee het kanon omhoog schiet: elevatie nul is een schot in horizontale richting, een elevatie van 90 graden is loodrecht omhoog.

```
1200 PRINT "beginsnelheid kogel";
1205 INPUT V
1210 PRINT "elevatie (0-90)";
1215 INPUT EL
```

In het vervolg van het programma rekenen we met het horizontale en het verticale bestanddeel van de snelheid (VH en VV). De beginwaarden daarvoor kunnen we nu halen uit de ingevoerde gegevens.

```
1217 EL=EL*3.14159/180:REM graden naar radialen
1220 VH=V*COS(EL):REM  beginsnelheid horizontaal
1230 VV=V*SIN(EL):REM  beginsnelheid verticaal
```

We gaan de gebeurtenissen van moment tot moment volgen. De tijdstap tussen die momenten is DT. Voor de totale duur van de gebeurtenissen nemen we ruim de tijd: TT=15. Aldus:

```
1300 DT=0.05:REM tijdstap
1310 TT=15:REM  totale duur
```

We gaan nu nauwkeurig kijken, wat er met de kogel gebeurt tijdens zijn vlucht.
In de verticale richting werken op de kogel twee krachten. De ene is de zwaarte-

kracht (FA), waarvan de grootte gelijk is aan de massa (M) maal de versnelling van de zwaartekracht (G). Vooraan in het programma moeten we deze grootheden eerst een waarde geven:

```
1110 G=10:REM  zwaartekrachtsversnelling
1120 M=1:REM  de kogelmassa
```

De zwaartekracht werkt naar beneden. Als we de snelheid omhoog als positief rekenen, moet hier een minteken staan.

```
2100 FA=-M*G:REM  kracht is gewicht
```

De tweede verticale kracht is de luchtweerstand FW. Die hangt af van de snelheid. We moeten daarbij goed uitkijken. Voor een kogel die zuiver horizontaal beweegt, is er geen verticale wrijvingskracht. De verticale wrijving hangt dus af van de verticale snelheid VV. De verticale wrijvingskracht is gericht tegen de verticale snelheid in. Dit geeft een minteken in de formule. Ook is de wrijving groter als de kogel snel beweegt, of dat nu horizontaal is of verticaal. Een snelle kogel komt meer lucht tegen, en die lucht biedt hem meer weerstand. Ook de totale snelheid VL ten opzichte van de lucht speelt dus een rol. We moeten met VL maal VL rekenen, want dat is een maat voor de energie van de lucht die de kogel op zich af ziet komen. Hoe we VL berekenen, zullen we later nog zien. We krijgen de formule:

```
2110 FW=-CW*VV*VL*VL:REM  verticale wrijving
```

Hierin komt een wrijvingsconstante CW voor, die bepaald is door de vorm van de kogel. We kiezen ervoor die constante zelf iedere keer in te voeren bij het begin van het programma.

```
1260 PRINT "wrijvingsconstante (0 of 1)";
1270 INPUT CW:REM  wrijvingsconstante
1280 CW=CW*.0001:REM  schaling
```

Dank zij regel 1280 hoeven we niet iedere keer een lastig rijtje nullen in te typen en behouden we toch de vrijheid CW elke gewenste waarde te geven. De totale verticale kracht FV is nu gelijk aan de zwaartekracht plus de verticale wrijvingskracht.

```
2120 FV=FW+FA:REM  totale verticale kracht
```

Volgens een van de wetten van Newton is de verticale versnelling gelijk aan de verticale kracht gedeeld door de massa, dus FV/M. Om te weten hoeveel de verticale snelheid na het tijdje DT is veranderd, moeten we de versnelling met dat tijdje DT vermenigvuldigen:

```
2130 V1=DT*FV/M:REM  verticale
                                snelheidsverandering
```

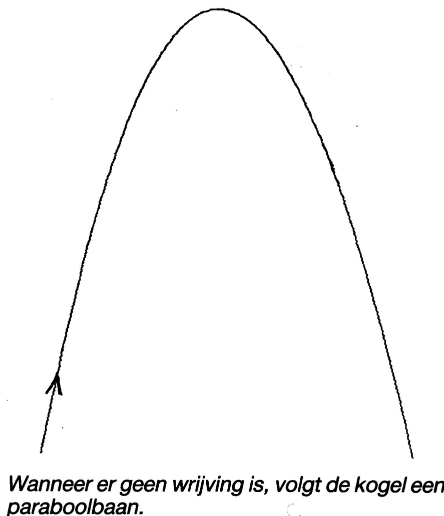
De verticale snelheid na die verandering is nu te berekenen uit de snelheid voordat de verandering plaatsvond:

```
2140 VV=VV+V1:REM  nieuwe snelheid verticaal
```

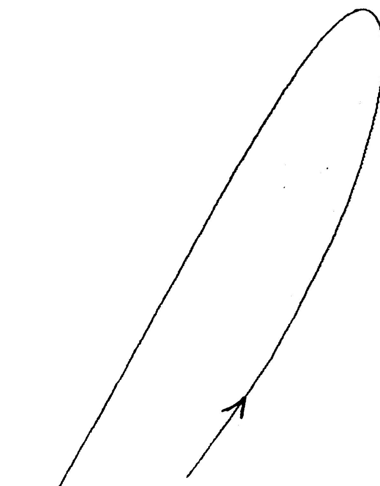
Tijdens de periode DT is de verticale snelheid gemiddeld iets kleiner dan deze nieuwe VV, namelijk VV-V1/2. Om te weten welke verticale afstand wordt afgelegd, moeten we dit vermenigvuldigen met de tijdstap DT. De totale verticale weg S1, afgelegd tot het einde van deze tijdstap, wordt dan:

```
2150 S1=S1+DT*(VV-V1/2):REM  verticale positie
```

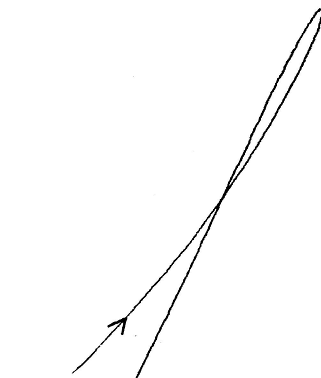
We moeten ervoor zorgen dat onze kogel op een hoogte nul van start gaat. Voorin het programma zetten we daarom:



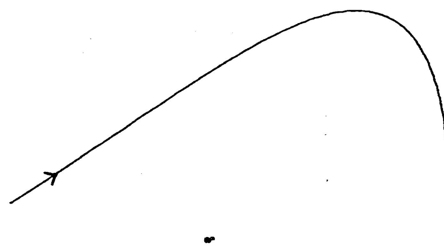
Wanneer er geen wrijving is, volgt de kogel een paraboolbaan.



Bij wrijving en tegenwind kan de kogel achter de eigen linies inslaan.



Bij bepaalde omstandigheden is het mogelijk dat een afgeschoten kogel zich gedraagd als een boemerang.



Bij tegenwind kan een kogel loodrecht van boven inslaan. In boeken uit vroeger eeuwen staan kogelbanen vaak zo getekend. Leerboeken die alleen eenvoudige paraboolbanen afleiden, waren er toen nog niet.

```
1320 S1=0:REM  verticale beginpositie
```

We stappen nu over op de horizontale richting. Daar volgen we hetzelfde principe. Eerst bekijken we de krachten, daarna de snelheid en ten slotte de afgelegde weg.
In horizontale richting is er geen zwaartekracht, wel luchtweerstand. Bij de luchtweerstand moeten we kijken naar de snelheid van de kogel ten opzichte van de lucht. Omdat er wind kan zijn, is dat niet zonder meer de snelheid van de kogel.
De windsnelheid moet eerst worden ingevoerd:

```
1250 PRINT "windsnelheid";
1255 INPUT VW
```

Een windsnelheid groter dan nul waait met de kogel mee (naar rechts). Willen we de kogel tegenwind geven, dan moeten we een negatief getal invoeren. Het horizontale bestanddeel L2 van de snelheid ten opzichte van de lucht is:

```
2020 L2=VH-VW:REM  tov wind, horizontaal
```

De totale snelheid ten opzichte van de lucht komt dan op:

```
2030 VL=SQR(L2*L2+VV*VV):REM  tov lucht, totaal
```

Op dezelfde manier als bij de verticale wrijvingskracht komen we tot de formules voor de horizontale wrijvingskracht FH. Die is tegelijk de totale horizontale kracht, waarmee we de nieuwe snelheid en de afgelegde weg kunnen berekenen.

```
2040 FH=-CW*L2*VL*VL:REM  horizontale wrijving
2050 V2=DT*FH/M:REM  horizontale
                          snelheidsverandering
2060 VH=VH+V2:REM  nieuwe snelheid horizontaal
2070 S2=S2+DT*(VH-V2/2):REM  horizontale positie
```

We mogen niet vergeten dat ook S2 een beginwaarde moet krijgen:

```
1330 S2=0:REM  horizontale beginpositie
```

Dit alles (horizontaal en verticaal) moet de computer telkens opnieuw uitrekenen. Dat gaat mooi met een FOR...NEXT lus. Dus:

```
2010 FOR T=0 TO TT STEP DT
3000 NEXT T:REM  volgende moment
```

Uitvoer programma

Tot nu toe hebben we nog niet nagedacht over de uitvoer die het programma moet geven. De opdrachten daarvoor moeten bij ieder stapje worden doorlopen en moeten dus komen voor regel 3000. Aan het begin van het programma moet het scherm echter eerst klaargezet worden voor grafische weergave.

```
1500 GOSUB 600:REM  grafisch scherm
1510 CN=0:REM  voorgrondkleur
```

Verder moeten we ervoor zorgen dat de punten die we gaan tekenen, binnen het scherm vallen. In Basicode-3 loopt de horizontale tekencoördinaat van 0 tot en met 1-1/HG; de verticale tekencoördinaat van 0 tot en met 1-1/VG. HG en VG zijn constanten van het systeem, die de aantallen beeldpunten op het scherm aangeven in horizontale en verticale richting. Dit zijn grote getallen en de coördinaten lopen dus ruwweg van 0 tot 1. In centimeters is dit bereik van de tekencoördinaat in verticale zin 3/4 van het totaal aantal centimeters in de breedte van het scherm. Als we

niets daartegen ondernemen, krijgen we dus een verkleining tot 75 procent in de verticale richting. Voor een getrouwe weergave moeten we in de horizontale richting de tekencoördinaat tot 75 procent verkleinen.
Dit is een ingewikkelde zaak. Gemakkelijker te aanvaarden is het feit dat de verticale tekencoördinaat VE nul is bovenaan het scherm en bijna 1 onderaan het scherm. Dat is precies omgekeerd aan de manier waarop we normaal met hoogte rekenen. In dit programma laten we een waarde S1=100 overeenkomen met de bovenkant van het scherm (VE=0). Om veranderen gemakkelijk te maken, brengen we dit getal 100 onder in een variabele VM. Het beginpunt, S1=0, moet 15 procent boven de onderrand van het scherm komen. Dat is bij VE=0.85. (Dat is vast en zeker groter dan 1-1/VG, want VG is minstens 72. VE=0.85 is dus op alle Basicode-3 computers een toegestane waarde.) Het getal 0.85 brengen we onder in een variabele BO (voor bodem).

```
1380 VM=100:REM  hoogte top scherm
1390 BO=.85:REM  tekencoordinaat
                 bodem
```

Nu gaan we de formules plaatsen voor het berekenen van de tekencoördinaten HO en VE.

```
2510 VE=BO*(1-S1/VM):REM
               tekenwaarde verticaal
2520 HO=(1-BO)+0.75*S2/VM:REM
              tekenwaarde horizontaal
```

De waarde 0.75 in regel 2520 is voor de getrouwe weergave. De 1-BO daar schuift het beginpunt ook 15 procent naar rechts. We zijn er nu nog steeds niet, want het kan heel goed voorkomen, dat de waarden van VE en HO buiten het scherm komen te vallen.

```
2530 NT=0:REM  wel tekenen
2540 IF VE<0 THEN NT=1:REM  te hoog
2550 IF HO<0 THEN NT=1:REM  te links
2560 IF VE>1-1/VG THEN NT=1:REM  te
                               laag
2570 IF HO>1-1/HG THEN NT=1:REM  te
                             rechts
2580 IF NT=1 THEN NG=1:GOTO 3000:REM
                     dit keer niets
```

De variabele NT staat voor niet tekenen. NG staat voor niet getekend. Deze dient om te onthouden dat het vorige punt niet getekend is. Er mag dan geen verbindingslijn gezet worden, wanneer het volgende punt aan de beurt komt. Een verbindingslijn mag ook niet gezet worden bij het eerste punt. Daarom:

```
1520 NG=1:REM  geen vorig punt
```

Eindelijk zijn we zover dat de eigenlijke tekenopdrachten kunnen komen. Dat er een punt getekend is, wordt onthouden voor de volgende keer.

```
2600 IF NG=1 THEN GOSUB 620:REM
                              punt
2610 IF NG=0 THEN GOSUB 630:REM
                           lijntje
2620 NG=0:REM  wel getekend
```

Nu nog even de afsluiting. Het programma gaat tamelijk lang door. We hebben alle gelegenheid de kogelbaan te bekijken. Het is daarom niet nodig het programma op het eind nog eens extra te laten wachten. We kunnen meteen terugspringen naar de invoer voor de volgende baan.

```
4990 GOSUB 100:REM  terug naar
                       tekstscherm
5000 GOTO 1200:REM  volgende invoer
```

Als we wel willen wachten op het moment dat de tekening klaar is en dan willen stoppen, is het alternatief:

```
4980 GOSUB 210:REM  wacht op toets
4990 GOTO 950:REM  stop
```

Als we willen afbreken, terwijl het programma nog aan de baan bezig is, kunnen we de volgende regels toevoegen. Een willekeurige toets brengt ons dan naar regel 4990. (Bij het alternatief naar regel 4980, waar nogmaals een toets moet worden aangeslagen.)

```
2400 GOSUB 200:REM  kijk of toets
2410 IF IN$<>"" THEN T=TT+1:REM
                        grens over
```

Misschien zijn er mensen die meerdere kogelbanen in één tekening willen hebben. Dat kan niet zonder meer, omdat INPUT niet kan plaatsvinden zolang we een grafisch scherm op hebben staan. Die mensen moeten dus vooral alle waarden regelen, waarvoor ze de baan willen zien. Het gaat hier om een Basicode-3 programma. Voorafgaand aan de regels 1000 en 1010 moeten dus de subroutines van Basicode-3 worden geplaatst. Die zijn te halen uit het Basicode-3 vertaalprogramma, dat in de boekhandel te koop is, of te verkrijgen bij de gebruikersclub van uw computer. Voor IBM-PC/MS-DOS computers staat een voorlopige versie van de subroutines in dit tijdschrift.
Na nog wat verfraaiingen komt ons totale programma er als volgt uit te zien. Zie onder.
Het is leuk te experimenteren met andere waarden van de beginsnelheid, de elevatie, de windsnelheid en eventueel de wrijvingsconstante en de massa. Let wel op de stapgrootte DT. Als die te klein wordt genomen, moeten we lang wachten om de kogel te zien bewegen. Wordt de stapgrootte te klein genomen, dan gaat de berekening fout. Als compromis hebben we DT=0.05 genomen.
Neem eerst de wrijvingsconstante nul. De wind heeft dan geen effect en de baan is de bergparabool die in de leerboeken wordt besproken. Geschikte beginsnelheden liggen dan tussen 10 en 40. Kijk eens bij telkens dezelfde beginsnelheid bij welke elevatie de kogel het verst komt. Voeg vervolgens wrijving toe, maar houdt de windsnelheid nog nul. Probeer daarna tegenwind (een negatief getal voor de windsnelheid). Neem bij veel tegenwind ook eens een kogel met een grotere massa.
Kleine veranderingen in de ingevoerde getallen kunnen soms het verschil uitmaken tussen een kogel die loodrecht op de vijand terecht komt en een kogel die achter de eigen linies inslaat. Probeer maar eens beginsnelheid 350, tegenwind -35 en wrijving 1. Laat de elevatie gaan van 30 naar 50 graden. In de militaire praktijk moet ook nog eens meetellen, dat de wind op verschillende hoogten verschillend is. Het is dus terecht dat de artillerie zowel weerballonnen als computers inschakelt voordat er geschoten wordt! □

```
1000 A=500:GOTO 20:REM  kogelbaan
1010 GOSUB 100:REM  scherm schoon
1020 PRINT "Dit programma tekent kogelbanen."
1030 PRINT
1110 G=10:REM  zwaartekrachtsversnelling
1120 M=1:REM  de kogelmassa
1200 PRINT "beginsnelheid kogel";
1205 INPUT V
1210 PRINT "elevatie (0-90)";
1215 INPUT EL
1217 EL=EL*3.14159/180:REM  graden naar radialen
1220 VH=V*COS(EL):REM  beginsnelheid horizontaal
1230 VV=V*SIN(EL):REM  beginsnelheid verticaal
1250 PRINT "windsnelheid";
1255 INPUT VW
1260 PRINT "wrijvingsconstante (0 of 1)";
1270 INPUT CW:REM  wrijvingsconstante
1280 CW=CW*.0001:REM  schaling
1300 DT=.05:REM  tijdstap
1310 TT=15:REM  totale duur
1320 S1=0:REM  verticale beginpositie
1330 S2=0:REM  horizontale beginpositie
1380 VM=100:REM  hoogte top scherm
1390 BO=.85:REM  tekencoordinaat bodem
1500 GOSUB 600:REM  grafisch scherm
1510 CN=0:REM  voorgrondkleur
1520 NG=1:REM  geen vorig punt
2010 FOR T=0 TO TT STEP DT
2020 L2=VH-VW:REM  tov wind, horizontaal
2030 VL=SQR(L2*L2+VV*VV):REM  tov lucht, totaal
2040 FH=-CW*L2*VL*VL:REM  horizontale wrijving
2050 V2=DT*FH/M:REM  horizontale snelheidsverandering
2060 VH=VH+V2:REM  nieuwe snelheid horizontaal
2070 S2=S2+DT*(VH-V2/2):REM  horizontale positie
2100 FA=-M*G:REM  kracht is gewicht
2110 FW=-CW*VV*VL*VL:REM  verticale wrijving
2120 FV=FW+FA:REM  totale verticale kracht
2130 V1=DT*FV/M:REM  verticale snelheidsverandering
2140 VV=VV+V1:REM  nieuwe snelheid verticaal
2150 S1=S1+DT*(VV-V1/2):REM  verticale positie
2400 GOSUB 200:REM  kijk of toets
2410 IF IN$<>"" THEN T=TT+1:REM  grens over
2510 VE=BO*(1-S1/VM):REM  tekenwaarde verticaal
2520 HO=(1-BO)+.75*S2/VM:REM  tekenwaarde horizontaal
2530 NT=0:REM  wel tekenen
2540 IF VE<0 THEN NT=1:REM  te hoog
2550 IF HO<0 THEN NT=1:REM  te links
2560 IF VE>1-1/VG THEN NT=1:REM  te laag
2570 IF HO>1-1/HG THEN NT=1:REM  te rechts
2580 IF NT=1 THEN NG=1:GOTO 3000:REM  dit keer niets
2600 IF NG=1 THEN GOSUB 620:REM  punt
2610 IF NG=0 THEN GOSUB 630:REM  lijntje
2620 NG=0:REM  wel getekend
3000 NEXT T:REM  volgende moment
4990 GOSUB 100:REM  terug naar tekstscherm
5000 GOTO 1200:REM  volgende invoer
32000 REM
```

UW SOFTWARE NU NOG VOORDELIGER !

Word Processing

Word Processing	DFL.
Display Write V	946,-
Multimate 3.13	650,-
PFS Professional Write	285,-
WordPerfect 4.2	525,-
Wordperfect 4.1 (Ned.)	**BEL !**
Wordstar 2000	637,-
Wordstar Pro Pak	595,-
XY Write III	545,-

Spreadsheets & integrated

Lotus 1-2-3	825,-
Lotus Hal	265,-
Lotus Report Writer	275,-
Symphony	1.142,-
PFS First Choice	235,-
Smart Integrated	1.125,-
VP Planner	145,-
MicroSoft Multiplan	295,-
Supercalc IV	**BEL !**

Database management

dBase III Plus	1.095,-
dBase III Lan Pak	1.495,-
Foxbase +	625,-
R:Base System V	895,-
Reflex	195,-

Utilities

Copy-II-pc	89,-
Disk Optimizer	89,-
Fixed Disk Organiser	145,-
Double Dos	89,-
Fastback	225,-
MicroSoft Windows	159,-
Sidekick (unprotected)	119,-
Superkey	98,-
Norton Utilities	125,-
Norton commander	89,-
Disk Optimizer	89,-

Graphics

Graphics	DFL.
Autocad Version 2.52	6.529,-
Chartmaster	595,-
GEM Graph	395,-
Lotus Freelance Plus	885,-
MicroSoft Chart	455,-
Prodesign II	475,-
Harvard Presentation	595,-
Freelance Plus	855,-

Languages & Tools

Clipper (dBase III compiler)	885,-
Lattice C compiler	650,-
Ryan McFarlan Fortran	825,-
Ryan McFarlan Cobol	1.295,-
MicroSoft Quick Basic compiler	175,-
MicroSoft Macro Assembler	235,-
Quicksilver dBase III compiler	895,-
Turbo Pascal	165,-
Turbo Prolog	165,-

Project Management

MicroSoft Project	595,-
Super Project Plus	735,-
Harvard Total Project	695,-
Timeline	575,-

Datacommunicatie

Crosstalk XVI	256,-
Remote	256,-
Smartcom II	235,-
Mirror	103,-

Ons leveringspakket is veel groter dan wij kunnen adverteren
Bel 030-626975 of na 15-8-1987 **613848 voor uitgebreide informatie.**

Vermelde prijzen zijn exclusief BTW. Orders boven fl.500,- franko thuis. Prijzen onder voorbehoud.

Tenzij anders vermeld, betreft het officiële U.S. pakketten.

SCA-*Soft*

Euterpedreef 8, 3561 CV Utrecht
Postbus 2277, 3500 GG Utrecht
Telefoon: 030 - 62 69 75
(na 15-08-1987) 030 - 613848

TWEE UNIEKE OPBOUWWERKEN VOOR COMMODORE BEZITTERS



Nog nooit werd in een boek zo diep ingegaan op alle facetten en mogelijkheden van de C64!

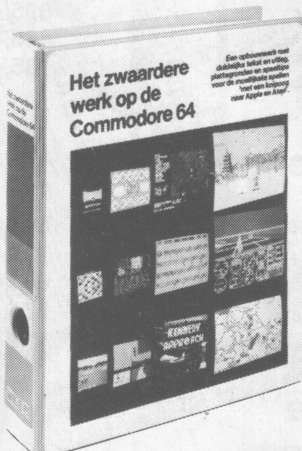
Stap voor stap leert u werken met

- Basic ● Hulptalen ● Geluid
- Graphics & Sprites ● Randapparatuur
- Machinetaal

Blijf niet meer steken in een eenvoudig spelletje!
Door de logische opbouw van de uitgave en de professionele voorbeelden komt u tot een volledig gebruik van uw computer en de randapparatuur.
Wegwijs worden in programmeertalen. Aan de hand van voorbeelden met praktisch te gebruiken subroutines leert u te werken niet alleen in BASIC maar ook in PASCAL, LOGO, PILOT, Simon's BASIC, ADA en minder bekende talen.
De grafische mogelijkheden en het geluid. vormen twee sterke kanten van de C 64. Een C 64 heeft 3 toongeneratoren waardoor de geluidskwaliteit van de machine bij velen geliefd is. En uiteraard zullen de meest bekende melodieën hier in de toekomst niet ontbreken.
Wat betreft de grafische mogelijkheden komen o.a. het juiste gebruik van o.a. **sprites** aan bod.
Nuttige adressen, produktinformatie, tips voor het gebruik van apparatuur en programmeren vormen de afsluiting van een werk dat beginner en gevorderde C 64-gebruiker van begin tot eind zal boeien.

GRATIS UNIEKE HULPMIDDELEN
Reeds in het basiswerk zijn enkele unieke hulpmiddelen opgenomen:
● "cheatsheets", bijvoorbeeld programmeer-commando's samengevat bij en op het toetsenbord. ● Computertekensjabloon, waarmee flowcharts, grafische software en sprites ontwikkeld kunnen worden.
● Papier voor grafisch ontwerpen.

EN OP KORTE TERMIJN:
Software op floppy en/of casette en speciale aanbiedingen.
Basiswerk: f 99,- (B.fr. 1800,-),
exclusief verzendkosten.
Aktualiseringen à f 55,- (B.fr. 1000,-).



In deze unieke uitgave wordt in **begrijpelijk Nederlands** een breed scala van de populaire maar moeilijke spellen voor de C 64 uitgelegd. Aan de hand van duidelijke beschrijvingen en afbeeldingen (schermfoto's) worden de spellen systematisch besproken. Door middel van trucs (praktische handigheden), speeltips en plattegronden dringt u door tot de essentie van het spel.

De volgende **categorieën van spellen** worden besproken:

- Simulatiespellen
- Avontuurspellen
- Denkspellen
- Beweeg- en schietspellen

Daarnaast wordt veel aandacht besteed aan de accessoires, die bij het spelen van spellen onmisbaar zijn.

Welke spellen kunt u zo al verwachten?
Een kleine greep:

- ULTIMA III;
- Ghostbusters;
- Natocommander;
- Cluedo;
- Summergames 1+2;
- Sky Fox;
- Bruce Lee;



En op het creatieve vlak, o.a.:

- Music Construction Set;
- Koala touchpad;
- Sketchpad.

In het basiswerk zijn ca 50 spellen opgenomen, in de aanvulling komen telkens circa 20 spellen.
Basiswerk f 99,- (B.fr. 1800,-),
exclusief verzendkosten.
Aanvullingen à f 55,- (B.fr. 1000,-).

Speciaal voor u
Bij deze uitgaven worden, waar nuttig en nodig, cheatsheets en cassettes met software bijgeleverd: zonder meerkosten.
De uitgaven worden geleverd in luxe A4 band (25 x 32 cm).

AANBIEDING
De ware Commodore enthousiast zal beide boeken willen hebben. Indien u nu besluit beide boeken tegelijk te bestellen betaalt u hiervoor geen f 198,- maar slechts f 158,- (B.fr. 2880,-).



WEKA Uitgeverij BV
Postbus 61196
1005 HD Amsterdam
tel. 020 - 86 71 31

BESTELBON

JA, zend mij direct/via boekhandel* de aangekruiste boeken toe. De aanvullingen à f 55,- ontvang ik automatisch tot wederopzegging.

☐ **Van Basic tot machinetaal op C 64** (f 99,- of B.fr. 1800,-)
☐ **Het zwaardere werk op C 64** (f 99,- of B.fr. 1800,-)
☐ **Beide uitgaven** tegen de speciale aanbiedingsprijs van f 158,- (B.fr. 2880,-).

Naam: ____________________
Adres: ____________________
PC/Plaats: ____________________
5223
Telefoon: ____________________
Datum: __________ Handtekening: __________
* Boekhandel: ____________________

Bon zenden aan: **Weka Uitgeverij B.V., Antwoordnummer 15412, 1000 PZ Amsterdam**

Kik Velt

Moderne bussen

In de vorige aflevering hebben we kennis gemaakt met de belangrijkste begrippen uit de wereld van de computerbussen. Tevens hebben we kennis gemaakt met een eenvoudig protocol, de synchrone bus. Dit keer zullen we ingewikkelder systemen gaan bekijken, de asynchrone bus en de verspreide bus. Doch eerst zullen we bekijken hoe die bussen er in de praktijk nu eigenlijk uitzien.

Deel 2 Busstandaards

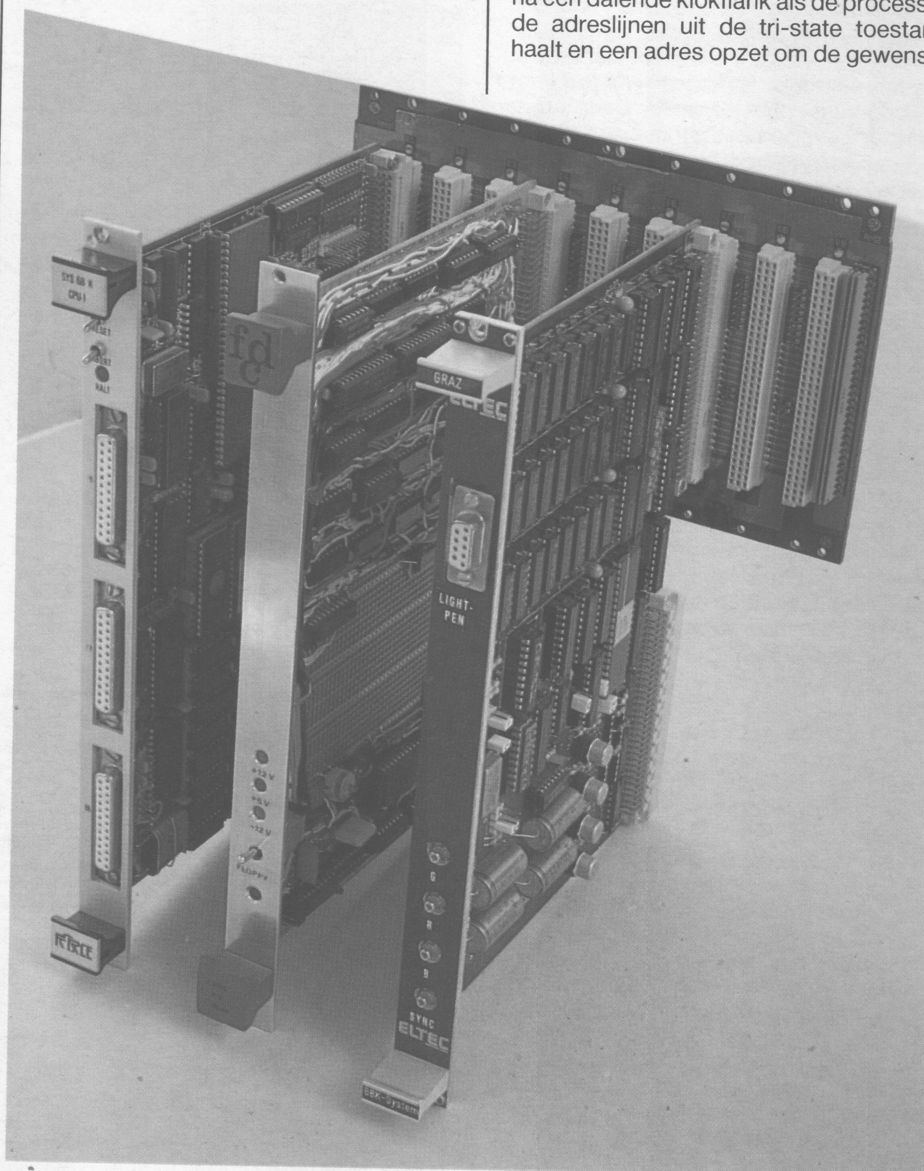
Connectoren

Tot nu zijn we vrij abstract bezig geweest met bussen. Maar hoe ziet een bus eruit? Een verzameling draadjes die kriskras over de printplaat lopen? Dat is inderdaad het geval, en als zodanig is een bus niet erg interessant. De ontwerper van het kaartje hoeft er alleen maar voor te zorgen dat de verbindingen juist gelegd zijn. Hoe zij precies lopen hoeft niemand te weten. Meestal echter blijkt het niet mogelijk te zijn alle gewenste IC's op een enkele printplaat te plaatsen, en moet de computer in meerdere kaarten worden opgesplitst. Deze kaarten moeten met draadjes aan elkaar verbonden worden omdat zij wel allemaal op dezelfde bus zitten. Draadjes zijn echter veel te lastig, liever gebruikt men hiervoor een aparte printplaat met connectoren waar de computerkaart precies inpast. Deze combinatie van connectoren en achterplaat wordt ook wel "de bus" genoemd. Het is inderdaad een duidelijk zichtbaar kenmerk van de bus, het enige echt zichtbare kenmerk zelfs, maar het is niet DE bus. Het is een onderdeel van het bussysteem.
Vroeger werden kaartrandconnectoren veelvuldig gebruikt. Vaak (maar lang niet altijd) kan men de gebruikte busstandaard herkennen aan de gebruikte connector. Tegenwoordig zijn de 64 of 96 polige DIN connectoren erg populair. Hun grote voordeel is dat de pinnetjes veel dichter op elkaar staan zodat met meer signalen gewerkt kan worden.

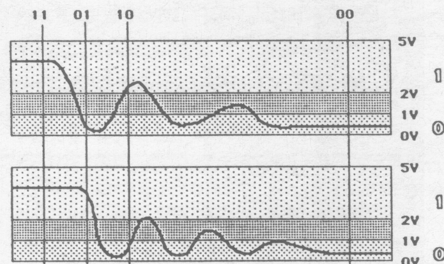
De asynchrone bus

Bij de synchrone bus, die we de vorige keer leerden kennen, werden alle handelingen op of binnen een bepaalde vaste tijd verricht. Iedereen, processor en geheugen, hield zich daar aan, en iedereen gehoorzaamde maar aan één enkel signaal: de klok.
Bij de asynchrone bus, de naam zegt het al, is dat niet het geval. Processor, geheugen, interfaces, zij communiceren nu in de handschud-mode met elkaar. Dat wil zeggen: via aparte signaallijntjes geeft de een aan de ander te kennen: "Ik ben daarmee of daarmee klaar, nu jij weer!"
Het is duidelijk dat als dit handschud protocol netjes verloopt elke communicatie zijn eigen snelheid bepaalt. Snelle geheugens zullen snel "OK" zeggen, tragere interfaces zullen even wachten. Maar niemand hoeft zich naar de langzaamste lummel te richten. Het nadeel is natuurlijk dat extra lijnen en voorzorgen noodzakelijk zijn.
Laten we als voorbeeld maar weer een eenvoudige leesopdracht bekijken van een typische asynchrone processor (zie afbeelding). Merk op dat ook hier een klok signaal aanwezig is. Maar deze klok heeft een totaal andere functie dan bij de synchrone bus. Hij is eigenlijk alleen ter referentie. De ene lijn mag niet voor de andere beginnen te schakelen. De cyclus begint na een dalende klokflank als de processor de adreslijnen uit de tri-state toestand haalt en een adres opzet om de gewenste

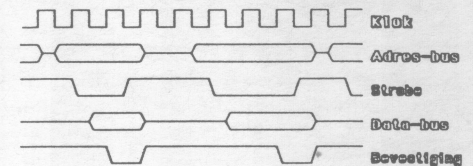
Voorbeeld van enkele computerkaarten in een moeder- of achterplaat. De eerste bevat de centrale procesoor met wat geheugen en I/O-interfaces. De 2e is een grafische interface naar een kleuren beeldscherm, en de 3e is een floppy-interface. Merk op dat kaarten van verschillende fabrikanten, ja zelfs eigen maaksel, zonder meer aan elkaar gekoppeld kunnen worden, omdat iedereen zich aan dezelfde busspecificatie houdt. ▼



byte te kunnen aanspreken. Zodra dit gebeurd is, wordt de zogenaamde strobe geactiveerd. Dat gebeurt bijvoorbeeld op de eerstvolgende stijgende klokflank, daar zit dan 50 ns tussen indien de klokfrequentie 10 MHz bedraagt. Dat activeren overigens houdt meestal in dat de lijn van 5V naar 0V wordt omgeschakeld, afhankelijk van het merk processor.
Nu zijn de diverse adresselectoren zo gemaakt dat zij pas reageren als de strobe actief geworden is. Tot die tijd, of zij hun adres nu wel of niet op de bus zagen staan, dienden zij zich koest te houden. De reden hiervoor is duidelijk. Bij de synchrone bus mochten de selectoren pas reageren bij de volgende klokflank en die was zodanig geplaatst dat de processor intussen alle adreslijnen allang hun juiste stand had gegeven. Maar bij de asynchrone bus kan dat niet en moet de strobe voor dit signaal zorgen. Immers, het omschakelen van de adreslijnen kost enige tijd, de lijnen denderen en bovendien veranderen zij door allerlei oorzaken niet allemaal tegelijk van waarde.
Stel dat tijdens een vorige cyclus 2 bepaalde adreslijnen allebei hoog (11) waren, en dan nu beide laag (00) moeten worden. Wie geeft de garantie dat de ene bit door welke oorzaak dan ook, niet iets eerder omslaat dan de ander? Dus dat de overgang via 01 of via 10 plaatsvindt? (Dit verschijnsel wordt "skew" genoemd). Kortom: de adres selectoren mogen pas op de adresstrobe reageren. Deze mag pas actief worden, nadat de adreslijnen tot rust zijn gekomen en zeker hun echte waarde hebben bereikt.
Ziet een selector zijn adres en tegelijk de strobe, dan mag de bijbehorende interface zijn data op de databus plaatsen. Daar kan hij lang òf kort over doen. Een snelle interface (bijvoorbeeld een RAM geheugen) zal de direct daarop volgende klokperiode klaar zijn. Hij kan dan de zogenaamde bevestigingslijn activeren hetgeen voor de processor het signaal is dat de data van de bus gelezen kan worden. Bij een trage interface (bijvoorbeeld een EPROM geheugen) zal meer tijd gevergd worden om de databus volledig aan te sturen. De systeemontwerper zal daarom een schakeling moeten toevoegen die de bevestiging 1, 2 of nog meer klokperiodes vertraagt zodat de verwerkingstijd evenredig verlengd wordt (zie figuur).
In alle gevallen wacht de processor op het bevestigingssignaal om de databus uit te lezen. Hij geeft het geheugen er alle tijd voor. Hij kan wachten tot in de eeuwigheid als het moet. En soms moet dat wel eens, bijvoorbeeld als de processor een stuk geheugen adresseert terwijl dat geheugen niet eens aanwezig is. Dan is de bevestigings-generator natuurlijk ook niet aanwezig, met duidelijke gevolgen vandien. Bij de synchrone bussen bestaat dat probleem niet. Na verloop van tijd leest de processor de databus uit, of er nu wel of niet iets zinnigs opgezet is. Maar bij asynchrone bussen moeten er speciale schakelingen toegevoegd worden die na verloop van tijd (enkele mikroseconden bijvoorbeeld) zeggen: "Nu hebben we lang genoeg gewacht, de bevestiging komt toch nooit meer, we zetten het alarm aan!" Het actief worden van de alarm lijn kan dan vervolgens gebruikt worden om de processor een alarm-routine, de zogenaamde bus-fout, te laten uitvoeren.
Maar het is niet de bedoeling dat het zo ver komt. Normaal gesproken krijgt de processor de bevestiging, leest de data van de bus en laat daarop de strobe los. Voor het geheugen op zijn beurt is dat weer het teken om zijn databus los te laten evenals het bevestigings-signaal. De processor reageert tenslotte daarop door de

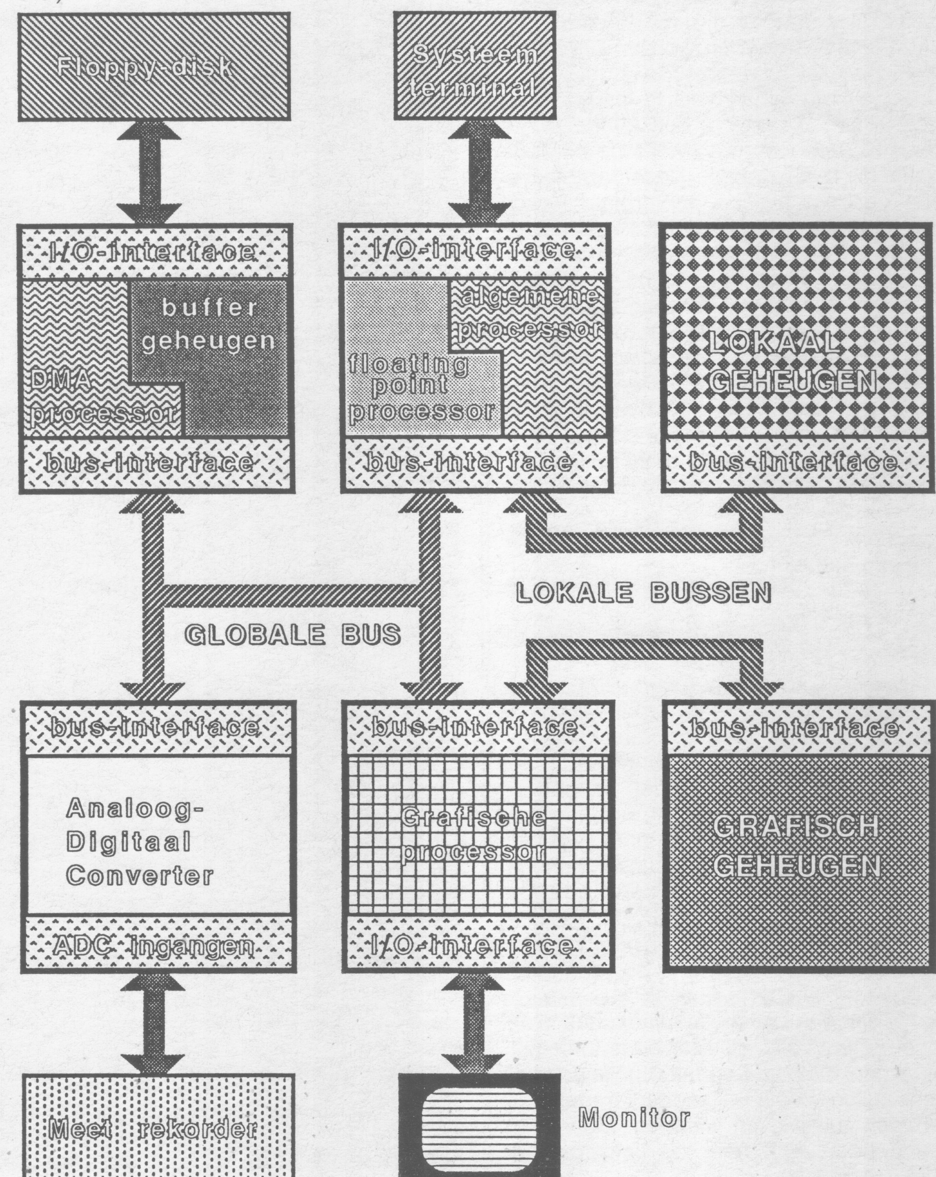


Afbeelding 2.
Twee elektrische signalen, allebei hoog (1) schakelen om naar laag (0). In werkelijkheid gaat dat niet zo gladjes en zwabberen de signalen vaak nog enkele tientallen nanoseconden op en neer. (Het zogenaamde "denderen".) Behalve de verwachte 11 en 00, worden ook de combinaties 01 en 10 eventjes aangetroffen! Verder is de logische toestand in het tussengebied (0.8V - 2.0V) ongedefinieerd, misschien 0, misschien 1.



Afbeelding 1.
Het typische tijdsdiagram van een asynchrone bus. (Vergelijk met de synchrone bus uit de vorige aflevering). Pas als de strobe aangeeft dat de adreslijnen OK zijn, mag er data overdracht plaatsvinden. Deze data overdracht wordt beëindigd door de bevestigingslijn omlaag te activeren. Wanneer een geheugen langere tijd nodig heeft om de databus te stabiliseren, stelt hij de bevestiging gewoon even uit en kan zo de cyclus vertragen.

Voorbeeld geavanceerd bussysteem (zie tekst).



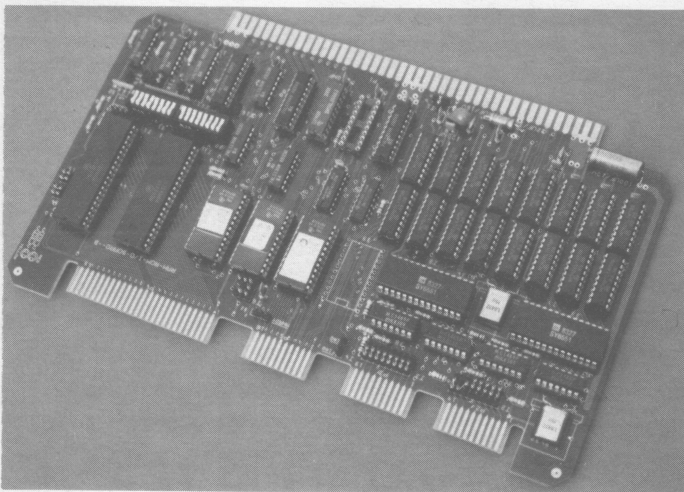
adresbus te tri-staten, waarmee de cyclus dus ten einde is.
De processor kan de ingelezen data nu intern gaan verwerken. De ene keer is hij daar snel mee klaar en kan direct de volgende dalende klokflank alweer een adres op de bus zetten. Een andere keer zal het wat langer duren en slaat hij een paar klokperiodes over. Bij de synchrone bus gaat dat ook zo. Alleen als de processor daar 1 periode moest overslaan was dat meteen 1000 ns en hier slechts 100. Ook dit werkt natuurlijk mee aan de veel grotere snelheid van een asynchrone processor boven een synchrone. Het nadeel is echter dat er meer lijnen nodig zijn. Het ontwerp ingewikkelder en dus duurder.

Multi processing

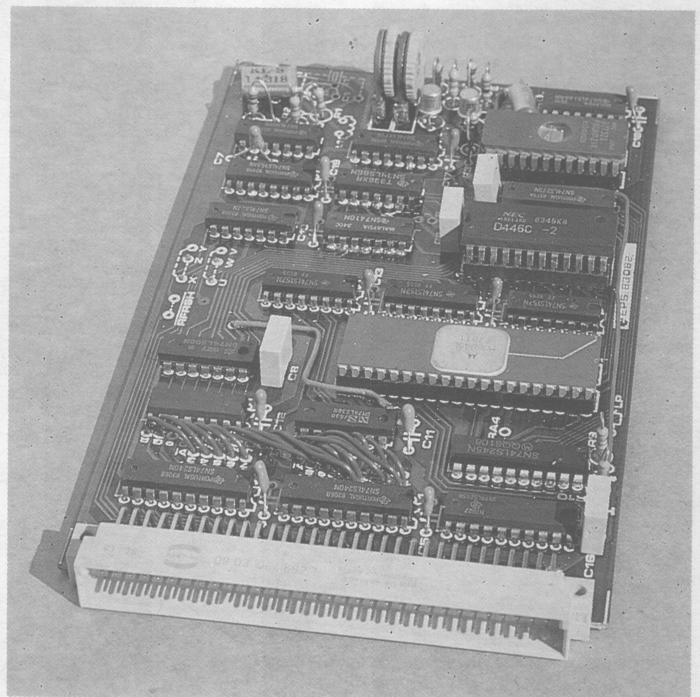
Tot nu toe zijn we uitgegaan van computersystemen met slechts 1 processor. Een ontwikkeling van eze tijd is echter dat de computers met meerdere processoren worden uitgevoerd zodat zij nog sneller en nog efficiënter worden. Deze multi-processor systemen kunnen erg simpel zijn: bijvoorbeeld een hoofdprocessor en een wat inferieurdere processor die alleen voor de beeldscherm uitvoer zorgt. Zij kunnen ook vrij complex zijn met een aantal nogal onafhankelijk werkende processoren die maar zo af en toe gegevens naar elkaar hoeven uit te wisselen. Dit noemt men een gedistribueerd systeem. Dit is een systeem waarin de intelligentie over verschillende processoren verdeeld is.
In alle gevallen kan echter nog maar steeds 1 processor tegelijk gebruik maken van de bus. Het is aan een speciale schakeling, de zogenaamde arbiter, om te bepalen welke processor toegang krijgt. De anderen zullen gedurende die tijd hun buslijnen in de tri-state toestand moeten houden. Zolang alle processoren vrij zelfstandig bezig zijn is dit alles geen bezwaar. Dat is bijvoorbeeld het geval wanneer zij gemiddeld 20% van de tijd de bus nodig hebben en maar met hun 3-en of 4-en zijn. Vervelender wordt het natuurlijk als zij bij elkaar meer dan 100% beslag leggen op de bus. In dat geval valt er met multi-processing niets meer te winnen, alleen nog maar te verliezen. Want de processoren brengen dan de meeste tijd met wachten door, wachten totdat zij van de arbiter op de bus mogen.

Lokale bus

Om in dat probleem weer te voorzien zal de systeemontwerper er voor moeten zorgen dat busgebruik zo min mogelijk optreedt en/of zo snel mogelijk gaat. Dat laatste kan gerealiseerd worden door een zo breed mogelijke databus te gebruiken (dus 32-bit en zeker geen 16-bit meer) en een zo hoog mogelijke werkfrequentie (minder dan 10 MHz gaat het vuilnisvat in). Minimalisatie van het busgebruik houdt in dat elke processor een duidelijk afgemeten taak toebedeeld krijgt en niet voor elk wissewasje gegevens van elders moet ophalen om te kunnen werken. Dit betekent dat zo een processor een apart stuk eigen geheugen moet hebben, zijn eigen in- en uitvoermedia, kortom dat hij eigenlijk een aparate computer begint te worden. De verschillende kaarten van deze computer zullen aan elkaar gekoppeld moeten worden, wat natuurlijk weer via een bus gaat. Zo ontstaat er een extra bus binnen de bus, een lokale bus. Elke processor kan op die manier aan een eigen bus zitten, een bus waar hij als enige wat over te zeggen heeft. En naast deze lokale bus is er een globale bus waar alle processoren samen aanzitten, doch nu onder aanvoering van de arbiter.
In de afbeelding zien we een mooi voorbeeld van een gedistribueerd bussysteem. Linksonder zien we een aparte ADC processor die de analoge signalen van een of ander meetinstrument moet verwerken. De voorbewerkte gegevens gaan over de hoofdbus naar de algemene processor (midden boven), die er allerlei bewerkingen mee kan gaan doen. Het uitfilteren van storingen en ruis in het signaal is zo een bewerking, en daarvoor kan op de kaart ook nog een aparte floating-point (reken) processor zitten. Bij deze berekeningen ontstaan vele tussenresultaten die even ergens moeten worden opgeslagen. Voor al die data is op de computerkaart wellicht geen plaats meer, daarvoor moet een aparte RAM kaart (rechtsboven) worden aangerukt. Deze wordt dan, om de hoofdbus niet nog extra te belasten op de lokale bus aangesloten. Voor latere analyse zal misschien data op schijf moeten worden opgeslagen, en dus zendt de algemene processor deze over de bus naar het buffergeheugen van de DMA processor (linksboven) die er verder dan maar zorg voor heeft te dragen dat de data veilig op schijf komt, of er vandaan wordt gelezen. Op een ander moment kan de algemene hoofdprocessor (die bij nader inzicht ook meteen de bus arbiter blijkt te zijn), besluiten data naar de grafische processor (midden onder) te zenden, die alle gegevens maar zo te verwerken heeft dat er een net plaatje op het beeldscherm ontstaat. Zo een grafisch systeem heeft ook veel geheugen nodig, en dus wordt er ook hier via een lokale bus nog een RAM kaart (rechtsonder) bijgekoppeld.
Tenslotte moet natuurlijk de gebruiker van



Voorbeeld van een eenvoudig computerkaartje met processor, klok, geheugen, interfaces. Duidelijk zijn de spoortjes van de gedrukte bedrading te zien. In werelijkheid overigens zijn de sporen meestal fotografisch ge-etst. Aan beide kanten zitten kaartrand connectoren voor de aansluiting op 2 verschillende byssystemen!



Voorbeeld van een modernere busaansluiting, de zogenaamde DIN connector. Hier een 2x32=64 polige connector. Indien meer signalen gewenst kan de middenrij worden opgevuld om tot 96 pinnen te komen.

het systeem ten alle tijden kunnen ingrijpen. Het is dus logisch dat er ergens een terminal zit aangesloten, minimaal een en wel op de kaart met de algemene processor. Maar om de afzonderlijke systemen ook afzonderlijk te kunnen testen, is het aanbevelenswaardig op alle vier de computerkaarten tenminste een afzonderlijke terminalaansluiting te hebben zitten. Natuurlijk had het systeem ook eenvoudiger en dus veel goedkoper gekund, met 1 enkele processor. Maar dan zou het natuurlijk wel een stuk trager zijn geweest. In dat geval zou je blij mogen zijn als je 100 metingen per seconde kunt verwerken, in plaats van de (vele) duizenden die je met dit systeem kunt halen.
Is dit het einde? Of zal een volgende generatie computersystemen alle arbiters onder een superarbiter en een superbus aan elkaar koppelen, zoals nu met alle processoren onder de arbiter gebeurd is? Wie zal het zeggen. Voorlopig is multi-processing (van industriële systemen) een ontwikkeling van nog maar een paar jaar oud. Sommige arbitersystemen moeten in 1987 nog op de markt komen. Voorlopig hebben de ingenieurs nog lang niet alle problemen opgelost die met de distributie samenhangen en de komende jaren zullen de technici er dan ook nog de handen aan vol hebbem. Pas als gedistribueerde systemen even lekker en soepel opereren als de mono-processor systemen nu, zal wellicht een nieuwe ontwikkeling op gang komen, maar dat kan nog wel enkele jaren duren.

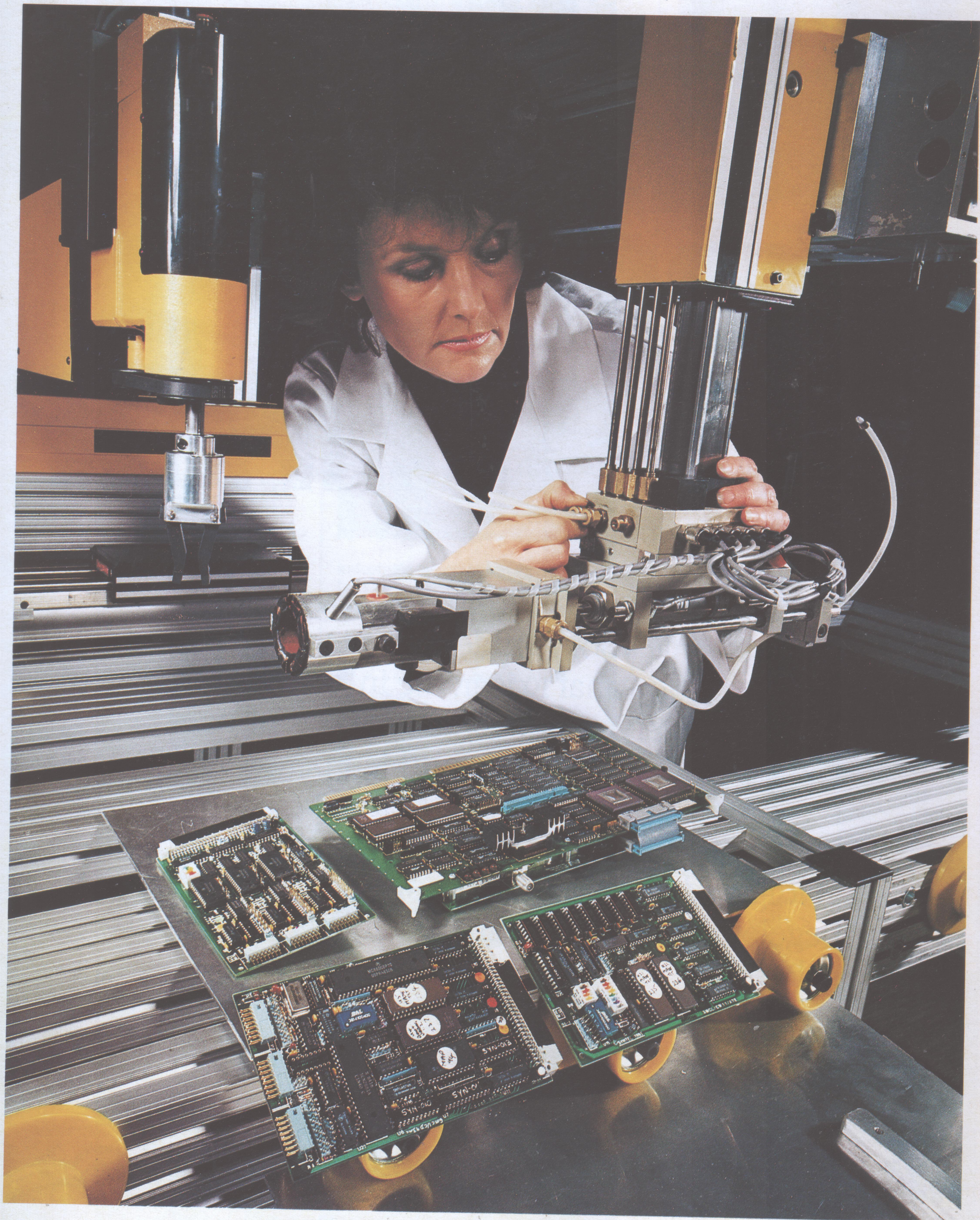
Evenwichtige krachtpatser ▷

Een robot die snelle bewegingen uitvoert met zware voorwerpen, heeft een goede kans door te schieten in zijn zwaai. Een ander probleem is doorbuiging van de arm. De klassieke oplossing is de arm erg stijf te maken. De arm wordt dan zelf ook wel erg zwaar.
De Tetrabot op de foto is een handiger oplossing. Hij is zo geconstrueerd dat het zwaartepunt meer naar het midden ligt. De reikwijdte van de Tetrabot is 1 meter. De maximale last is 10 kilogram, terwijl toch een nauwkeurigheid van 1/20 millimeter wordt gehaald. De Tetrabot wordt op het ogenblik nog verder ontwikkeld door GEC Research Ltd. in Engeland. De toepassingen liggen onder andere in de auto-industrie. (W.v.T.)



Persluchtrobot

Er bestaan zoemende robots, aangedreven door elektromotoren, en briesende robots, aangedreven door perslucht. Het apparaat op de foto is van het laatste soort. Het is een produkt van Norgren Martonair, een Engelse fabrikant van machines met persluchtaandrijving. De robot is opgebouwd uit standaardonderdelen. Uit die standaardonderdelen kunnen allerlei robots worden samengesteld, die dan precies zijn toegesneden op de gewenste toepassing. De nauwkeurigheid van de verplaatsingen van deze robot is een halve milimeter, terwijl de maximale bewegingssnelheid 1 meter per seconde bedraagt. (W.v.T.)

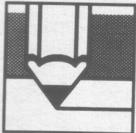



KINDERKAARTJE
VOOR 1 GULDEN
RAILRUNNER

UW COMMODORE 64 KAN MEER!

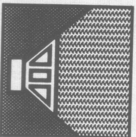
LEER IN ZES MAANDEN DRIE VAN ZIJN SPECIFIEKE TOEPASSINGEN BEHEERSEN

Met de cursus Commodore 64-expert van de Leidse Onderwijsinstellingen kunt u uw Commodore 64 tot op de laatste chip leren kennen! De cursus is samengesteld volgens een geheel nieuw concept: u bepaalt zelf in welke toepassingen u zich wilt verdiepen. U hebt daarbij de keuze uit zowel zakelijk als hobbyistisch gerichte specialisaties. Bij iedere module hoort software op cassette en diskette (tenzij anders aangegeven).



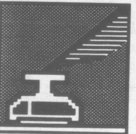
GRAFISCHE MOGELIJKHEDEN

LOW-RES, HI-RES, sprites, karakters etc. Met ontwerpblok en een serie programma's: "Sprite Machine" Graphic BASIC etc.



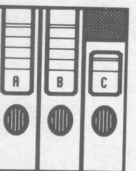
MUZIEK EN GELUIDSEFFECTEN

De SID-chip, ASDR, synchronisatie, ringmodulatie, filtering etc. Met o.a. programma "Music Studio".



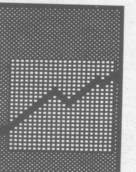
TEKSTVERWERKING

Manipulatie met teksten, briefindelingen, tabelwerk, "mail merge", etc. Bijbehorend tekstverwerkingspakket: Aackotext.



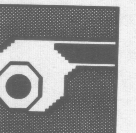
DATABASE ("ELEKTRONISCHE KAARTENBAK")

Met behulp van het programma "Databasis" ervaart u wat het opzetten van en werken met bestanden inhoudt.



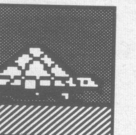
SPREADSHEET ("ELEKTRONISCH REKENBLAD")

Financiële berekeningen: hypotheken, produktie- en autokostenberekening etc. Met programma "Autocalc 64".



PROGRAMMEREN IN MACHINETAAL

Het hart van de C-64, de 6510-microprocessor, geheel ontbloot. Inclusief HK-Supercombitool cartridge (insteekmodule).



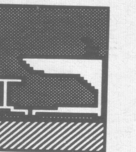
PROGRAMMEREN IN LOGO (LCN-LOGO)

Met uniek LCN-LOGO-cartridge (128K-systeem!) leert u de beginselen van LOGO kennen: de schildpad, lijsten etc.



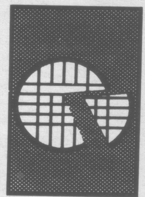
LESGEVEN EN COMPUTERS/ TOEPASSINGEN ALGEMEEN

Leer de fijne kneepjes van computergebruik in het onderwijs. Compleet met software-beoordelingsformulieren en programma's (listings).



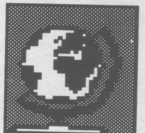
LESGEVEN EN COMPUTERS/ BASISONDERWIJS

Zet de computer in als typemachine of binnen het vak wereldoriëntatie. Met acht programma's, direct in de klas te gebruiken.



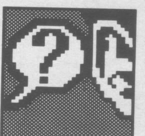
LESGEVEN EN COMPUTERS/ V.O. (ECONOMIE)

Leer hoe u de computer in de economieles kunt gebruiken. Met o.a. ondernemers-simulatiespel en zes programma's.



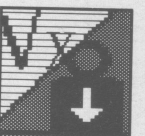
LESGEVEN EN COMPUTERS/ V.O. (AARDRIJKSKUNDE)

De computer is meer dan een "veredelde" overheadprojector! Met behulp van de zes programma's maakt u kennis met simulaties, leerspelen enz.



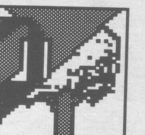
LESGEVEN EN COMPUTERS/ V.O. (TALEN)

Schakel de computer in bij tekstreconstructie, luisteroefeningen of taalspelletjes. Inclusief zes programma's.



LESGEVEN EN COMPUTERS/ V.O. (WIS- EN NATUURKUNDE)

Leer de computer te gebruiken bij het opstellen van vergelijkingen, tekenen en grafieken etc. Met zeven programma's.



LESGEVEN EN COMPUTERS/ V.O. (BIOLOGIE EN SCHEIKUNDE)

Hoe verloopt het proces van koolstofassimilatie? Leg deze en andere vragen uit met behulp van een computer. Zeven voorbeeldprogramma's.

Het aantal toepassingen (modulen) dat u bestudeert bepaalt u zelf, evenals de volgorde waarin u deze bestudeert. Het studieprogramma stelt u dus zelf op, naar eigen voorkeur. Uit de in totaal 14 modulen kiest u minimaal drie modulen, dat is al een complete cursus. De cursus duurt dan zes maanden. Vier, vijf, zes of meer modulen volgen kan ook.

WILT U MEER WETEN?

Vraag dan de gratis studiegids aan. Vul de bon in of bel ons even: 071-451882.

Stuur mij gratis en vrijblijvend de studiegids over de cursus Commodore 64-expert:

dhr./mw.: ______________________

straat: ______________________

postcode/woonplaats: ______________________

____

In een open envelop zonder postzegel sturen naar: LOI, Antwoordnummer 1, 2300 VB Leiden.

leidse onderwijs instellingen

1075 0

Erkend door de minister van onderwijs en wetenschappen bij beschikking van 5 maart 1975, kenmerk BVO/SFO-129.718. Leidsedreef 2, 2352 BA Leiden

Speciale aanbieding voor de lezers van "Aarde&Kosmos-DJO"

Minerals of the world

Een in prachtige kleuren uitgevoerde wandkaart van maar liefst 86 x 136 cm waarop 200 mineralen zijn afgebeeld. Kompleet met mineralogische, kristallografische, chemische en natuurkundige gegevens.

Speciaal voor scholen, studenten, amateurs, verzamelaars, hobbyisten een iedereen met belangstelling voor mineralen.

Deze unieke kaart maakt het mogelijk om heel snel en eenvoudig mineralen te herkennen met bijbehorende gegevens. Een Nederlandse tekstbegeleiding is bijgevoegd.

Deze wandkaart kost normaal 30 gulden. Voor u als lezer van ,,Aarde&Kosmos/DJO'' slechts 24,95 inklusief de verzendkosten (de kaart wordt opgerold in een koker verzonden).

Extra korting bij meer exemplaren:

2 tot 5 stuks -10%
6 tot 10 stuks -15%
11 tot 20 stuks -20%.

Meer dan 20 exemplaren: op aanvraag.

Bestellen door overmaking van het verschuldigde bedrag op giro 4998215 tnv de stichting Mens en Wetenschap te Huizen-Nh.



86 x 138 cm

SATELLIETKAART van Nederland

Sinds 1972 wordt ons land regelmatig gefotografeerd doorLandsat-kunstmanen. Uit vier opnamen, gemaakt op 1 en 2 november, is nu een groot formaat fotokaart in vier kleuren samengesteld, waarop Nederland en België tot de lijn die over Luik en Brussel loopt, te zien zijn, zonder dat er één wolkje boven het land hangt. De kaart is geproduceerd door het ITC en het NLR. Er is een nieuwe bewerkingstechniek gebruikt die kleuren heeft opgeleverd die dichter bij de werkelijkheid komen dan de ,,valse-kleuren'' die we gewoonlijk op Landsat-opnamen zien.



De kaart meet 94 x 123 cm en bezit een schaal van 1:275.000.Door het grote formaat konden zeer veel details in de opnamen weergegeven worden.
De kaart is uitgevoerd op zwaar papier, gevat in twee metalen rails waardoor hij minder kwetsbaar en makkelijk kan worden opgehangen.
De kaart is opgerold en verpakt in een stevige koker. Er zit een toelichtend boekje van 16 pagina's bij.

De kaart kan besteld worden onder nummer 80-56. De prijs is 49,50 (inklusief de verzendkosten).
Bestellen door storting van het verschuldigde op giro 4998215 tnv de stichting Mens en Wetenschap te Huizen-Nh.